Applied and
Computational
Mathematics
Division

NISTIR 89-4103

Center for Computing and Applied Mathematics

User's Reference Guide for ODRPACK: Software for Weighted Orthogonal Distance Regression Version 1.7

*P.T. Boggs, R.H. Byrd,
J.R. Donaldson, and R.B. Schnabel*

August 1989

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Gaithersburg, MD 20899

User's Reference Guide

for

ODRPACK
Software for Weighted Orthogonal Distance Regression

Version 1.71
07-27-89

National Institute of Standards and Technology
(formerly National Bureau of Standards)
Internal Report 89-4103

Paul T. Boggs
Applied and Computational Mathematics Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

Richard H. Byrd
Department of Computer Science
University of Colorado
Boulder, CO 80309

Janet R. Donaldson
Applied and Computational Mathematics Division
National Institute of Standards and Technology
Boulder, CO 80303-3328

Robert B. Schnabel
Department of Computer Science
University of Colorado
Boulder, CO 80309
and
Applied and Computational Mathematics Division
National Institute of Standards and Technology
Boulder, CO 80303-3328

keywords
orthogonal distance regression
measurement error models
nonlinear least squares
errors in variables

categories
G2E,I1B1

*** REVISION HISTORY ***

Revision	Description
1.71 (07-27-89)	ODRPACK 1.71 corrects an error in the code that performs the computation of finite difference derivatives when M>=2 and the default value of IFIXX is invoked. (The default value of IFIXX is invoked when IFIXX(1,1) is set to a negative value or when ODRPACK routines DODR or SODR are called.) This error could result in incorrect ''fixing'' of the independent variables, which would affect the final solution. Such ''fixing'' could be detected by observing the presence of values of DELTA that were identically zero. The error could go undetected by the user, however, if the values of DELTA were not examined after the fit.

TABLE OF CONTENTS

I. INTRODUCTION

ODRPACK is a portable collection of ANSI 77 Fortran subroutines for fitting a model to data. It is designed primarily for instances when the independent as well as the dependent variables have significant errors, implementing a highly efficient algorithm for solving the weighted orthogonal distance regression problem, i.e., for minimizing the sum of the squares of the weighted orthogonal distances between each data point and the curve described by the model equation. It can also be used to solve the ordinary least squares problem where all of the errors are attributed to the observations of the dependent variable. A complete description of the orthogonal distance regression problem and the algorithm implemented in ODRPACK is given by Boggs et al. [1987a and 1987b].

ODRPACK is designed to handle many levels of user sophistication and problem difficulty.

* It is easy to use, providing two levels of user control of the computations, extensive error handling facilities, optional printed reports and no size restrictions other than effective machine size.

* The necessary derivatives (Jacobian matrices) are approximated numerically if they are not supplied by the user.

* The correctness of user supplied derivatives can be verified by the derivative checking procedure provided.

* Both weighted and unweighted analysis can be performed.

* Subsets of the unknowns can be treated as constants with their values held fixed at their input values, allowing the user to examine the results obtained by estimating subsets of the unknowns of a general model without rewriting the model subroutine.

* The covariance matrix and the standard errors of the model parameter estimators are optionally provided.

* The ODRPACK scaling algorithm automatically compensates for poorly scaled problems, in which the model parameters and/or unknown errors in the independent variables vary widely in magnitude.

* It can accommodate complex data and multiple response data, i.e., data where the dependent variable is multi-dimensional. (See section III.)

* The trust region Levenberg-Marquardt algorithm implemented by ODRPACK has a computational effort per step that is of the same order as that required for ordinary least squares, even though the number of unknowns estimated in the orthogonal distance regression problem is the number of unknown model parameters plus the number of independent variables, while the number of unknowns estimated in the ordinary least squares problem is simply the number of unknown model parameters.

* The code is portable and is easily used with other Fortran subroutine libraries.

The following sections describe ODRPACK in greater detail. Users are directed to section II for a brief description of the orthogonal distance regression algorithm. This section introduces notation and provides background material for understanding the remainder of the documentation.

Section III describes how ODRPACK can be used for complex and multiple response data, and is only required for users with these data types. Section IV describes the need for starting values for BETA and DELTA, and section V describes two features of ODRPACK that simplify the user interface with the package. The information in these two sections will be especially important to first time users of ODRPACK. The subroutine declaration and call statements are given in section VI and the subroutine arguments are defined in section VII. The sample programs shown in section VIII can be used as templates for creating the user's own program. The information provided in section IX describes the scaling algorithm and section X describes how the user can extract computed results from the work vectors. The information in these two sections is generally not needed by first time users of ODRPACK.

II. BACKGROUND

Let

Y(I) = FN(X(I,*)+DELTA(I,*);BETA) - EPSILON(I) (eq.1)

for I=1,...,N, where

N
is the number of observations (see subroutine argument N);

Y(I), I=1,...,N
are the observed values of the dependent variable, where Y(I) depends on X(I,J), J=1,...,M (see subroutine argument Y);

FN
is the function used to predict values of the dependent variable (see subroutine argument FUN);

X(I,J), I=1,...,N & J=1,...,M
are the observed values of the independent variable (see subroutine argument X);

DELTA(I,J), I=1,...,N & J=1,...,M
are the unknown errors in X(I,J) that are to be estimated (see subroutine arguments JOB and WORK);

BETA(K), K=1,...,NP
are the function parameters that are to be estimated (see subroutine argument BETA);

EPSILON(I), I=1,...,N
are the unknown errors in Y(I) that are to be estimated (see subroutine argument WORK).

We are assuming that

```
observed< Y(I) >    = true< Y(I) >   - true< EPSILON(I) >
observed< X(I,J) >  = true< X(I,J) > - true< DELTA(I,J) >
```

and thus that

```
estimated< Y(I) >   = observed< Y(I) >   + estimated< EPSILON(I) >
estimated< X(I,J) > = observed< X(I,J) > + estimated< DELTA(I,J) >.
```

The square of the weighted orthogonal distance from the point (X(I,*),Y(I)) to the point FN(X(I,*)+DELTA(I,*);BETA) on the curve described by the model equation, i.e., the square of the observation errors, is given by

```
R(I)**2 = [FN(X(I,*)+DELTA(I,*);BETA) - Y(I)]**2
                                                                        (eq.2)
             M
          + SUM [D(I,J)*DELTA(I,J)]**2
            J=1
```

for I = 1,...,N, where

D(I,J), I=1,...,N & J=1,...,M — are the DELTA weights, which can be used to compensate for instances when the precision of the X observations is different from that of the Y observations (see subroutine argument WD).

The least squares orthogonal distance solution is then that which minimizes with respect to BETA and DELTA the weighted sum of the squared observation errors,

```
 N
SUM [ ( W(I)  * R(I)  )**2 ]                                            (eq.3)
I=1
```

where

W(I), I=1,...,N — are the observation error weights, which can be used to compensate for unequal precision in the observation errors, R(I) (see subroutine argument W).

The solution is found using a trust region Levenberg-Marquardt method [Boggs et al., 1987b], with scaling used to accommodate problems in which estimated values have widely varying magnitudes. The Jacobian matrices, i.e., the matrices of first partial derivatives of FN with respect to each BETA and each X, are computed at every iteration either by finite differences or by a user supplied subroutine, as specified by subroutine argument JOB (see section VII.B). The iterations are stopped when any one of three stopping criteria are met. Two of these indicate the iterations have converged to a solution. These are "sum of squares convergence", which indicates that the change in the weighted sum of the squared observation errors is sufficiently small, and "parameter convergence", which indicates the change in the values of BETA and DELTA is sufficiently small. The third stopping criteria is a limit on the number of iterations.

III. MULTIPLE RESPONSE DATA

Since its initial release, users have been interested in applying ODRPACK to complex data and to multiple response data in general. Although ODRPACK was written for single response data, where only one dependent variable is observed for each independent variable, it is possible to use it to handle multiple response data, where the dependent variable is multi-dimensional. Complex dependent data falls under the category of multiple response data since the real and imaginary parts of the dependent variable must be treated as separate observations.

Let Y(I,L), L=1,...,Q be the Q responses for the Ith observation of the independent variable, X(I,*). These Q multiple responses of the dependent variable cannot simply be treated as Q separate observations as can be done for ordinary least squares because ODRPACK would then treat the independent variables associated with these Q observations as unrelated and thus not constrain the errors DELTA(I,*) to be the same for each of the Q occurrences of X(I,*). In the multiple response case, therefore, the square of the observation errors (eq.2) must be defined as

```
            Q
R(I)**2 = SUM  (C(I,L)*[FN(X(I,*)+DELTA(I,*);BETA) - Y(I,L)])**2
          L=1
                                                           (eq.4)
              M
          + SUM [D(I,J)*DELTA(I,J)]**2
              J=1
```

for I = 1,...,N, where

FN(X(I,*)+DELTA(I,*);BETA)-Y(I,L) is the estimated error in the Lth response of the Ith observation of the dependent variable, and

C(I,L), I=1,...,N & L=1,...,Q must be appropriately chosen based on the desired weights for the individual response functions.

Equation (eq.4) has the effect of collapsing the Q errors associated with Y(I,L),L=1,...,Q, into a single value. This implies that NP must be less than or equal to N, rather than less than or equal to N*Q as would be the case if the multiple response problem were handled directly by ODRPACK or the problem were solved using ordinary least squares. Future plans for ODRPACK include modifications that will allow multiple response data to be handled directly, thus eliminating this restriction.

ODRPACK actually computes the results specified by (eq.2) using

$$R(I)**2 = [\langle F(I)\rangle - \langle Y(I)\rangle]**2 + \sum_{J=1}^{M} [D(I,J)*DELTA(I,J)]**2 \qquad (eq.5)$$

for I = 1,...,N, where

<F(I)> is the value in the Ith location of vector F returned from the user supplied subroutine FUN, which in the single response case contains FN(X(I,*)+DELTA(I,*);BETA); and

<Y(I)> is the value supplied in the Ith location of vector Y of the ODRPACK subroutine argument list, which in the single response case contains the Ith observation of the dependent variable.

ODRPACK can thus be "tricked" into solving multiple response orthogonal distance regression problems by setting

$$\langle F(I)\rangle = sqrt(\sum_{L=1}^{Q} (C(I,L)*[FN(X(I,*)+DELTA(I,*);BETA) - Y(I,L)])**2) \qquad (eq.6)$$

for I = 1,...,N, within user supplied subroutine FUN, and setting

<Y(I)> = 0.0

for I = 1,...,N. The computations specified by (eq.5) will then yield the value specified by (eq.4) and the multiple response ODR problem will be solved correctly.

Note that this technique for solving multi-response orthogonal distance regression problems has the advantage of retaining the original size of the problem, i.e., NP parameters and N observations. It has the disadvantage, however, of making the function F a more complicated function of BETA and DELTA than the original function FN. For small data sets, therefore, users may want to consider explicitly including each DELTA(I,J) as part of an expanded vector BETA and solving the resulting (NP+N*M) parameter problem using ordinary least squares as described in Boggs and Donaldson [1989] or Fuller [1987].

Note also that the standard errors of a multi-response orthogonal distance regression problem encoded as shown in (eq.6) will not be the same as those obtained by solving the problem as an ordinary least squares problem with (NP+N*M) parameters because the two functions being minimized have different Jacobian matrices at the solution. (See Section VII.B, subroutine argument JOB and IPRINT.)

IV. STARTING VALUES FOR BETA AND DELTA

Starting values for BETA must be provided by the user. Users familiar with the ordinary nonlinear least squares problem are generally aware of the importance of obtaining good starting values for the estimated function parameters. It is equally important to obtain good starting values for the parameters when using the orthogonal distance regression technique. Good starting values can significantly decrease the number of iterations required to find a solution; a poor starting value may even prevent the solution from being found at all. Reasonable starting values are often available from previous analysis or experiments. When good starting values are not readily available, the user may have to do some preliminary analysis to obtain them. Himmelblau [1970] offers several suggestions for obtaining starting values when they are not available from other sources.

When using the technique of orthogonal distance regression it is also important to have good starting values for the estimated errors, DELTA, in the independent variables. The ODRPACK default is to initialize DELTA to zero, which is the most obvious initial value for the DELTAs. (Note that zero starting values for DELTA do not cause the scaling problems discussed in section VII.B that zero starting values for BETA cause.) Initializing the DELTAs to zero, however, is equivalent to initially assigning all of the errors to the dependent variable as is done for ordinary least squares. While initializing the DELTAs to zero is quite adequate in many cases, in others it is not. A plot of the curve described by the model function and observed data for the initial parameters may indicate whether or not zero starting values for DELTA are reasonable. Often it is visually possible to determine better starting values for the DELTAs, especially when an asymptote is involved. For example, in the case of an asymptote, the user may need to initialize some of the DELTAs to the horizontal distance to the curve, while leaving the other DELTAs initialized to zero in order to obtain a reasonable solution. This problem is discussed further in [Boggs et al., 1987b]. As noted there, proper initialization of DELTA can mean the difference between solving a difficult problem and not solving it.

V. DEFAULT VALUES AND STRUCTURED ARGUMENTS

ODRPACK uses default values and structured arguments to simplify the user interface. The availability of default values in ODRPACK means that the user does not have to be concerned with determining values for many of the ODRPACK arguments unless the problem being solved requires the use of nondefault values. Structured arguments, which exploit the possibly symmetric structure of the independent variable data, reduce the amount of storage space required for arguments and reduce the work required by the user to initialize those arguments.

DEFAULT VALUES. Default values have been specified for ODRPACK subroutine arguments wherever feasible. These default values are invoked by setting the argument to any negative value. Arrays with default values are invoked by setting the first element of the array to a negative value, in which case only the first value of the array will ever be used. This allows a scalar to be used to invoke the default values of arrays, thus saving space and the need to declare such arrays.

Users are encouraged to invoke the default values of arguments wherever possible. The default values have been found to be reasonable for a wide class of problems. Their use will greatly simplify the initial use of ODRPACK for a given problem. Fine tuning of these arguments can then be done later if it is found necessary.

STRUCTURED ARGUMENTS. Structured arguments are included in ODRPACK because the properties of the individual elements of the possibly multiple column independent variable data are often constant throughout a given column of the independent variable or even throughout the whole independent variable matrix. For example, section II introduces the DELTA weights, specified by subroutine argument WD, that indicate how the DELTA and EPSILON for each observation (X,Y) are to be weighted in the weighted orthogonal distance. If each row of the independent variable indicates an hourly temperature reading and each column a different day on which the temperature readings were taken, then the user would probably want to weight each of the DELTAs equally. If one column of the independent variable contained hourly temperature readings and the other hourly humidity readings, then the user might want to weight each of the DELTAs in the first column the same, and to weight each of the DELTAs in the second column the same, but not necessarily want to weight the two columns equally. Of course, in other cases, the user might want to weight each of the DELTAs differently.

ODRPACK structured arguments exploit this possible symmetry as follows. If each of the N by M elements of an array describing some property of the independent variable are identically equal, then a single value can be used to specify all N by M elements. If the values of such an array only vary between columns, then each column of the array can be specified by a single value. Thus, it is only necessary to supply all N by M elements of the structured argument array when the elements of one or more of the columns must be individually specified.

The use of ODRPACK structured arguments is summarized as follows.

Structure of property P to be specified by structured argument A	Encoding of structured argument A and its leading dimension LDA	Accessed elements of structured argument A	Resulting assignment of property P
Property P constant throughout independent variable matrix	A(1,1) < zero with LDA = 1 or LDA >= N	A(1,1)	P(I,J) = -A(1,1), I=1,...,N & J=1,...,M
Property P varies only between columns of independent variable matrix	A(1,1) >= zero with LDA = 1	A(1,J), J=1,...,M	P(I,J) = A(1,J), I=1,...,N & J=1,...,M
Property P varies between and within columns of independent variable matrix	A(1,1) >= zero with LDA >= N	A(I,J), I=1,...,N & J=1,...,M	P(I,J) = A(I,J), I=1,...,N & J=1,...,M

If the first element of the structured argument is negative, then each of the N by M elements described by the argument is set to the absolute value of the first element. In this case, only the first element of the structured argument is ever referenced, allowing the user to set the N by M elements using only a scalar. (Note that in this case, setting the first element to a negative value does not necessarily invoke a default value.) This feature thus saves space and the need to declare the structured argument as an array.

If the first element of the structured argument is positive, then the way the structured argument will be used to designate the N by M values specified by it will depend its leading dimension. The leading dimension of the structured argument can be either exactly equal to one, or greater than or equal to N. When the leading dimension is exactly equal to one, the structured argument must be passed to ODRPACK as a one by M row vector containing the M values used to set each of the M columns. When the leading dimension is greater than or equal to N, the structured argument passed to ODRPACK must contain an N by M array of values.

VI. SUBROUTINE DECLARATION AND CALL STATEMENTS

The declaration and call statements for ODRPACK's user callable routines, SODR, SODRC, DODR and DODRC, are given below. SODR and SODRC invoke the single precision version of the code and DODR and DODRC invoke the double precision version. SODR and DODR preset many arguments to their default values and therefore have shorter call statements than SODRC and DODRC. SODRC and DODRC have expanded call statements that give the user greater control in solving the orthogonal distance regression problem. The information in this section is provided primarily for reference. Users are directed to section VII for example programs. These examples, which use Fortran PARAMETER statements to dimension ODRPACK arrays, provide a recommended format for creating an ODRPACK driver that will allow future changes to be made easily.

Note that although ODRPACK is distributed in both single precision and double precision versions, both versions may not be available to the user. In addition, even when both versions are available, the single precision version may not be appropriate to use. This is because ODRPACK is sensitive to the machine's precision, and requires approximately 14 decimal places. Somewhat fewer places should still work, but six or seven decimal places are definitely too few for general use, since only the simplest problems could be solved correctly at such reduced precisions.

When both versions are available, the user must choose which version of ODRPACK to use based upon which version supplies adequate precision on the target machine. To our knowledge, at present only Cray and CDC machines offer sufficient precision to permit general use of the single precision version of ODRPACK. For other machines, we recommend the double precision version.

If both versions of ODRPACK have sufficient precision on the user's machine, then either may be used. When both the single and double precision versions are available, however, there are trade offs between them. The double precision version will offer greater accuracy in results, while the single precision version will require less storage and possibly less machine time.

SODR: Compute the weighted orthogonal distance regression or ordinary linear or nonlinear least squares solution in single precision. (SODR is appropriate for general use only on machines with approximately 14 decimal places of precision for single precision.) Derivatives are either supplied by the user or numerically approximated by ODRPACK. Control values are preset, and a three part report of the results can be optionally generated.

```
      PROGRAM MAIN
       .
       .
       .
      EXTERNAL
     +   FUN,JAC
      INTEGER
     +   N,M,NP,
     +   LDX,
     +   LDWD,
     +   JOB,
     +   IPRINT,LUNERR,LUNRPT,
     +   LWORK,IWORK(LIWORK),LIWORK,
     +   INFO
      REAL
     +   X(LDX,M),
     +   Y(N),
     +   BETA(NP),
     +   WD(LDWD,M),
     +   WORK(LWORK)
       .
       .
       .
      CALL SODR
     +   (FUN,JAC,
     +   N,M,NP,
     +   X,LDX,
     +   Y,
     +   BETA,
     +   WD,LDWD,
     +   JOB,
     +   IPRINT,LUNERR,LUNRPT,
     +   WORK,LWORK,IWORK,LIWORK,
     +   INFO)
       .
       .
       .
      END
```

SODRC: Compute the weighted orthogonal distance regression or ordinary linear or nonlinear least squares solution in single precision. (SODRC is appropriate for general use only on machines with approximately 14 decimal places of precision for single precision.) Derivatives are either supplied by the user or numerically approximated by ODRPACK. Control values are supplied by the user, and a three part report of the results can be optionally generated.

```
      PROGRAM MAIN
       .
       .
       .
      EXTERNAL
     +   FUN,JAC
      INTEGER
     +   N,M,NP,
     +   LDX,IFIXX(LDIFX,M),LDIFX,LDSCLD,
     +   IFIXB(NP),
     +   LDWD,
     +   JOB,NDIGIT,
     +   MAXIT,
     +   IPRINT,LUNRPT,LUNERR,
     +   LWORK,IWORK(LIWORK),LIWORK,
     +   INFO
      REAL
     +   X(LDX,M),SCLD(LDSCLD,M),
     +   Y(N),
     +   BETA(NP),SCLB(NP),
     +   WD(LDWD,M),W(N),
     +   TAUFAC,
     +   SSTOL,PARTOL,
     +   WORK(LWORK)
       .
       .
       .
      CALL SODRC
     +   (FUN,JAC,
     +   N,M,NP,
     +   X,LDX,IFIXX,LDIFX,SCLD,LDSCLD,
     +   Y,
     +   BETA,IFIXB,SCLB,
     +   WD,LDWD,W,
     +   JOB,NDIGIT,TAUFAC,
     +   SSTOL,PARTOL,MAXIT,
     +   IPRINT,LUNERR,LUNRPT,
     +   WORK,LWORK,IWORK,LIWORK,
     +   INFO)
       .
       .
       .
      END
```

DODR: Compute the weighted orthogonal distance regression or ordinary linear or nonlinear least squares solution in double precision. Derivatives are either supplied by the user or numerically approximated by ODRPACK. Control values are preset, and a three part report of the results can be optionally generated.

```
      PROGRAM MAIN
      .
      .
      .
      EXTERNAL
     +   FUN,JAC
      INTEGER
     +   N,M,NP,
     +   LDX,
     +   LDWD,
     +   JOB,
     +   IPRINT,LUNERR,LUNRPT,
     +   LWORK,IWORK(LIWORK),LIWORK,
     +   INFO
      DOUBLE PRECISION
     +   X(LDX,M),
     +   Y(N),
     +   BETA(NP),
     +   WD(LDWD,M),
     +   WORK(LWORK)
      .
      .
      .
      CALL DODR
     +   (FUN,JAC,
     +   N,M,NP,
     +   X,LDX,
     +   Y,
     +   BETA,
     +   WD,LDWD,
     +   JOB,
     +   IPRINT,LUNERR,LUNRPT,
     +   WORK,LWORK,IWORK,LIWORK,
     +   INFO)
      .
      .
      .
      END
```

DODRC: Compute the weighted orthogonal distance regression or ordinary linear or nonlinear least squares solution in double precision. Derivatives are either supplied by the user or numerically approximated by ODRPACK. Control values are supplied by the user, and a three part report of the results can be optionally generated.

```
      PROGRAM MAIN
      .
      .
      .
      EXTERNAL
     +   FUN,JAC
      INTEGER
     +   N,M,NP,
     +   LDX,IFIXX(LDIFX,M),LDIFX,LDSCLD,
     +   IFIXB(NP),
     +   LDWD,
     +   JOB,NDIGIT,
     +   MAXIT,
     +   IPRINT,LUNRPT,LUNERR,
     +   LWORK,IWORK(LIWORK),LIWORK,
     +   INFO
      DOUBLE PRECISION
     +   X(LDX,M),SCLD(LDSCLD,M),
     +   Y(N),
     +   BETA(NP),SCLB(NP),
     +   WD(LDWD,M),W(N),
     +   TAUFAC,
     +   SSTOL,PARTOL,
     +   WORK(LWORK)
      .
      .
      .
      CALL DODRC
     +   (FUN,JAC,
     +   N,M,NP,
     +   X,LDX,IFIXX,LDIFX,SCLD,LDSCLD,
     +   Y,
     +   BETA,IFIXB,SCLB,
     +   WD,LDWD,W,
     +   JOB,NDIGIT,TAUFAC,
     +   SSTOL,PARTOL,MAXIT,
     +   IPRINT,LUNERR,LUNRPT,
     +   WORK,LWORK,IWORK,LIWORK,
     +   INFO)
      .
      .
      .
      END
```

VII. SUBROUTINE ARGUMENT DESCRIPTIONS

VII.A Synopsis

The arguments of the ODRPACK user callable subroutines are logically grouped as shown below. Arguments shown in parenthesis (...) are not included in the SODR and DODR call statements; SODR and DODR automatically preset these variables to the default values given in section VII.B. All other arguments are common to all ODRPACK user callable subroutines.

Argument Number	Arguments	Group Description
1 to 2	FUN,JAC,	Names of user supplied subroutines for function and Jacobian matrix computation
3 to 5	N,M,NP,	Problem size specification
6 to 11	X,LDX,(IFIXX,LDIFX,SCLD,LDSCLD,)	Independent variable information
12	Y,	Dependent variable
13 to 15	BETA,(IFIXB,SCLB,)	Function parameter information
16 to 18	WD,LDWD,(W,)	Weights
19 to 21	JOB,(NDIGIT,TAUFAC,)	Computation and initialization control
22 to 24	(SSTOL,PARTOL,MAXIT,)	Stopping criteria
25 to 27	IPRINT,LUNERR,LUNRPT,	Print control
28 to 31	WORK,LWORK,IWORK,LIWORK,	Work vectors and returned results
32	INFO	Stopping condition

VII.B Detailed Descriptions of ODRPACK User Callable Subroutine Arguments

The arguments of ODRPACK's user callable subroutines are described below in order of their occurrence in the call statements. Appropriate declaration statements for each argument are shown in brackets [...] following the argument name; the character string <real> denotes REAL when using single precision subroutines SODR and SODRC, which should be used only on machines with approximately 14 decimal digits of precision in single precision, and denotes DOUBLE PRECISION when using double precision subroutines DODR and DODRC. Each argument is numbered as shown in section VII.A, allowing the user to easily find the definition of a specific argument. In addition, three common characteristics of ODRPACK subroutine arguments are flagged in the left margin by the argument number. The flags are:

C which indicates the argument is only included in the call statements for SODRC and DODRC (SODR and DODR will preset these variables to their default values);

D which indicates the argument has a default value that can be invoked by setting the argument to any negative value; and

S which indicates the argument exploits possible symmetry in the properties of the independent variables as described in section IV.

NOTE

Substitute REAL for <real> when using SODR and SODRC.

Substitute DOUBLE PRECISION for <real> when using DODR and DODRC.

****

1. FUN [EXTERNAL FUN]

The name of the user supplied subroutine that computes the predicted values, F, of the dependent variable given the current values of the independent variable, XPLUSD=X+DELTA, and the function parameters, BETA. The subroutine argument list and declaration statements must be exactly as shown below.

```
      SUBROUTINE FUN(N,NP,M,BETA,XPLUSD,LDXPD,F,ISTOPF)
C
C  INPUT ARGUMENTS
C  (WHICH MUST NOT BE CHANGED BY THIS ROUTINE)
C
      INTEGER N,NP,M,LDXPD
      <real> BETA(NP),XPLUSD(LDXPD,M)
C
C  OUTPUT ARGUMENTS
C
      INTEGER ISTOPF
      <real> F(N)

      < computations for F(I)=FN(XPLUSD;BETA), I=1,...,N >

      < set ISTOPF = 0 if the current estimates of BETA and XPLUSD >
      <                   were acceptable for use in subroutine FUN>
      <                   and the regression procedure should      >
      <                   continue                                 >
      <             > 0 if the current estimates of BETA and XPLUSD >
      <                   were not acceptable for use in subroutine>
      <                   FUN, and values closer to the most       >
      <                   recently tried acceptable values of BETA >
      <                   and XPLUSD should be used                >
      <             < 0 if the regression procedure should be      >
      <                   stopped immediately                      >

      RETURN
      END
```

where

INTEGER N
is the number of observations, i.e., the number of points (X,Y).

INTEGER NP
is the number of function parameters, i.e., the number of values in vector BETA.

INTEGER M
is the number of columns of data in the independent variable matrix XPLUSD.

<real> BETA(NP)
is the singly subscripted array that contains the current values of the NP function parameters.

<real> XPLUSD(LDXPD,M)
is the doubly subscripted array that contains the current value of the N by M matrix of the independent variables, i.e., XPLUSD = X + DELTA.

INTEGER LDXPD
is the leading dimension of array XPLUSD.

<real> F(N)
is the singly subscripted array that contains the N predicted values of the function given the current values of the function

parameters and the independent variables, i.e., F = FN(XPLUSD;BETA).

INTEGER ISTOPF

is an indicator value that can be used to reject the current estimates of BETA and XPLUSD as unacceptable. Upon return from subroutine FUN:

If ISTOPF = 0 then

the current estimates of BETA and XPLUSD were acceptable for use in subroutine FUN, and the values of the predicted values F were properly computed. The regression procedure will continue.

If ISTOPF > 0 then

the values of the predicted values F could not be properly computed because the current estimates of BETA and XPLUSD were not acceptable. The regression procedure will select values closer to the most recently tried acceptable values of BETA and XPLUSD.

If ISTOPF < 0 then

the regression procedure should be stopped immediately. The final summary of the computation report will be printed, however, if it has been requested (see argument IPRINT).

2. JAC [EXTERNAL JAC]

The name of the user supplied subroutine that computes the Jacobian matrices, i.e., the matrices of first partial derivatives of FN with respect to each BETA and each X. This subroutine must be supplied only when digit C of JOB is nonzero (see subroutine argument JOB) although the external statement must always be provided in the user's main program; when digit C of JOB is zero the necessary Jacobian matrices will be computed by ODRPACK using finite differences.

Note that the logical argument ISODR, which is passed to subroutine JAC by ODRPACK, can be used to avoid computing the Jacobian matrix with respect to X when the fit is by the method of ordinary least squares and these derivatives are not needed. ISODR will be "false" in this case. It is not an error to compute the Jacobian with respect to X when the fit is by the method of ordinary least squares; it is an error if the Jacobian with respect to X is not computed when the fit is by the method of orthogonal distance regression.

The subroutine argument list and dimension statements must be exactly as shown below.

```
      SUBROUTINE JAC(N,NP,M,BETA,XPLUSD,LDXPD,
     +               FJACB,LDFJB,ISODR,FJACX,LDFJX,ISTOPJ)
C
C  INPUT ARGUMENTS
C  (WHICH MUST NOT BE CHANGED BY THIS ROUTINE)
C
      INTEGER N,NP,M,LDXPD
      LOGICAL ISODR
      <real> BETA(NP),XPLUSD(LDXPD,M)
C
C  OUTPUT ARGUMENTS
C
      INTEGER ISTOPJ
      <real> FJACB(LDFJB,NP),FJACX(LDFJX,M)

      < computations for FJACB(I,K)=first partial derivative of FN
                                    with respect to BETA(K),
                                    K=1,...,NP, at each observation
                                    I=1,...,N >

      IF (ISODR) THEN

      < computations for FJACX(I,J)=first partial derivative of FN
                                    with respect to X(I,J),
                                    J=1,...,M at each observation
                                    I=1,...,N >

      END IF

      < set ISTOPJ =  0 if the current estimates of BETA and XPLUSD>
      <                 were acceptable for use in subroutine   >
      <                 JAC and the regression procedure should >
      <                 continue                                >
      < set ISTOPJ <> 0 if the regression procedure should be   >
      <                 stopped immediately                     >

      RETURN
      END
```

where

INTEGER N
   is the number of observations, i.e., the number of points (X,Y).
INTEGER NP
   is the number of function parameters, i.e., the number of values in vector BETA.
INTEGER M
   is the number of columns of data in the independent variable matrix XPLUSD.
<real> BETA(NP)
   is the singly subscripted array that contains the current values of the NP function parameters.
<real> XPLUSD(LDXPD,M)
   is the doubly subscripted array that contains the current value of the N by M matrix of the independent variables, i.e., XPLUSD = X + DELTA.
INTEGER LDXPD
   is the leading dimension of array XPLUSD.
<real> FJACB(LDFJB,NP)
   is the doubly subscripted array that contains the N by NP matrix

of derivatives with respect to BETA at the current values of the function parameters and the independent variables.

INTEGER LDFJB

is the leading dimension of array FJACB.

LOGICAL ISODR

is a control value that can be used to inhibit the computation of the derivatives with respect to X when the solution is being computed by ordinary least squares and the derivatives with respect to X are not needed.

```
If ISODR is true then
   the solution is being computed by ODR and the derivatives with
   respect to X must be computed
else
   the solution is being computed by OLS and the derivatives with
   respect to X are not needed.
```

<real> FJACX(LDFJX,M)

is the doubly subscripted array that contains the N by M matrix of derivatives with respect to X at the current values of the function parameters and the independent variables, needed only when ISODR = true.

INTEGER LDFJX

is the leading dimension of array FJACX.

INTEGER ISTOPJ

is an indicator value that can be used to reject the current estimates of BETA and XPLUSD as unacceptable. Upon return from subroutine JAC:

```
If ISTOPJ = 0 then
   the current estimates of BETA and XPLUSD were acceptable for
   use in subroutine JAC, and the Jacobians were properly
   computed.  The regression procedure will continue.
Else
   the regression procedure should be stopped immediately.  The
   final summary of the computation report will be printed,
   however, if it has been requested (see argument IPRINT).
```

3. N [INTEGER N]

The number of observations, i.e., the number of points (X,Y). (See subroutine arguments X and Y.)

4. M [INTEGER M]

The number of columns of data in the independent variable matrix X. (See subroutine argument X.)

5. NP [INTEGER NP]

The number of function parameters, i.e., the number of values in vector BETA. (See subroutine argument BETA).

6. X [<real> X(LDX,M)]

The doubly subscripted array that contains the observed values of the N by M matrix of independent variables.

7. LDX [INTEGER LDX]

The leading dimension of array X.

LDX must equal or exceed N; values of LDX less than N will be treated as an input error.

CDS 8. IFIXX [INTEGER IFIXX(LDIFX,M)]

The doubly subscripted array that contains the indicator values used to designate whether element X(I,J), I=1,...,N & J=1,...,M, of the independent variable matrix is to treated as without error, i.e., DELTA(I,J) is to be fixed at zero, or whether the error DELTA(I,J) in that observation of the independent variable is to be estimated.

By default, all of the independent variables are treated as "unfixed", i.e. the errors DELTA(I,J) are estimated for all I=1,...,N & J=1,...,M. The default value is invoked when IFIXX(1,1) is set to any negative value. Other options for specifying IFIXX are described below.

```
If IFIXX(1,1) >= 0 then

   the way IFIXX is used depends on the value of the leading
   dimension of IFIXX, i.e., on LDIFX.

   If LDIFX = 1 then

      IFIXX must contain a 1 by M matrix of values, where for
      J=1,...,M

         if IFIXX(1,J) = 0 then

            X(I,J), I=1,...,N, is treated as exact and
            DELTA(I,J), I=1,...,N, is fixed at zero

         else

            X(I,J), I=1,...,N, is treated as approximate and
            DELTA(I,J), I=1,...,N, is estimated.

   If LDIFX >= N then

      IFIXX must contain an N by M matrix of values, where for
      I=1,...,N & J=1,...,M

         if IFIXX(I,J) = 0 then

            X(I,J) is treated as exact and DELTA(I,J) is fixed at
            zero

         else

            X(I,J) is treated as approximate and DELTA(I,J) is
            estimated.

If IFIXX(1,1) < 0 then

   the default option is invoked, i.e., each observation of the
```

independent variable, X(I,J), is treated as being measured with error an DELTA(I,J) that is estimated as described above in section II. In this case, only the first element of IFIXX is ever referenced and IFIXX can be a scalar.

C 9. LDIFX [INTEGER LDIFX]

The leading dimension of array IFIXX.

LDIFX must exactly equal one or must equal or exceed N; values of LDIFX less than one or between two and N-1, inclusive, will be treated as an input error.

See subroutine argument IFIXX for further details.

CDS 10. SCLD [<real> SCLD(LDSCLD,M)]

The doubly subscripted array that contains the scale values of the errors in the independent variable, i.e., the reciprocals of the expected magnitudes or typical sizes of DELTA(I,J), I=1,...,N & J=1,...,M.

Scaling is used within the regression procedure in order that the units of the variable space will have approximately the same magnitude. In particular, the scale value times the corresponding value of DELTA should be approximately one. For example, if DELTA(1,1) is expected to lie between -10E10 and 10E10 then SCLD(1,1) should be set to 10E-10, while if DELTA(1,1) is expected to lie between -10E-2 and -10E-4 then SCLD(1,1) should be set to 10E3. (The reciprocal of the standard errors of the observation X(I,J) can be used as SCLD(I,J) if the standard errors are known.) Except as noted in the next paragraph, the scale values specified for each DELTA must be greater than zero; values less than or equal to zero will be treated as an input error.

By default, the scale values will be set using the algorithm given in section IX.B. The default values are invoked when SCLD(1,1) is set to any negative value. Other options for specifying SCLD are described below.

If SCLD(1,1) > 0 then

each value of SCLD must be greater than zero and the way SCLD is used depends on the value of the leading dimension of SCLD, i.e., on LDSCLD.

If LDSCLD = 1 then

SCLD must contain a 1 by M matrix of values, and the scale of DELTA(I,J), I=1,...,N, is set to SCLD(1,J) for J=1,...,M.

If LDSCLD >= N then

SCLD must contain an N by M matrix of values, and the scale of DELTA(I,J) is set to SCLD(I,J) for I=1,...,N & J=1,...,M.

If SCLD(1,1) <= 0 then

the default option is invoked and each DELTA(I,J) is scaled as described in section IX.B. In this case, only the first element of SCLD is ever referenced and SCLD can be a scalar.

C11. LDSCLD [INTEGER LDSCLD]

The leading dimension of array SCLD.

LDSCLD must exactly equal one or must equal or exceed N; values of LDSCLD less than one or between two and N-1, inclusive, will be treated as an input error.

See subroutine argument SCLD for further details.

12. Y [<real> Y(N)]

The singly subscripted array that contains the N observed values of the dependent variable. (See section III for a discussion of how to handle multiple response data.)

13. BETA [<real> BETA(NP)]

The singly subscripted array that contains the (current) values of the NP function parameters.

On input: BETA must contain initial approximations for the function parameters. Initial approximations should be chosen with care since poor initial approximations can significantly increase the number of iterations required to find a solution and possibly prevent the solution from being found at all.

Users who do not provide scale information are strongly encouraged not to use zero as an initial approximation since a zero value can result in incorrect scale value selection by the scaling algorithm (see section IX). Setting the initial approximation to the largest magnitude that, for the user's problem, is effectively zero rather than the actual value zero will eliminate scaling problems, possibly producing faster convergence. For example, if BETA(1) represents change in cost in millions of dollars, then the value 10.0 might be considered "effectively zero", while if BETA(1) represents the change in cost in tens of dollars, then the value 0.01 might be considered "effectively zero."

On return: BETA contains the "best" estimate of the solution at the time the computations stopped.

CD 14. IFIXB [INTEGER IFIXB(NP)]

The singly subscripted array that contains the indicator values used to designate whether the corresponding value in BETA is to be treated as a fixed constant or is to be estimated.

By default, all of the function parameters, BETA, are treated as "unfixed", i.e. each of the BETA(K), K=1,...,NP, is estimated. The default value is invoked when IFIXB(1) is set to any negative value. Other options for specifying IFIXB are described below.

```
If IFIXB(1) >= 0 then

   IFIXB must contain a vector of NP values, where for K=1,...,NP

      if IFIXB(K) = 0 then

         BETA(K) will be held fixed at its input value

      else

         BETA(K) will be estimated as described above.

If IFIXB(1) < 0 then

   the default option is invoked, i.e., all BETA(K), K=1,...,NP,
   will be estimated as described above in section II.  In this
   case, only the first element of IFIXB is ever referenced and
   IFIXB can be a scalar.
```

CD 15. SCLB [<real> SCLB(NP)]

The singly subscripted array that contains the scale values of the function parameters, i.e., the reciprocals of the expected magnitudes or typical sizes of BETA(K), K=1,...,NP.

Scaling is used within the regression procedure in order that the units of the variable space will have approximately the same magnitude. In particular, the scale value times the corresponding value of BETA should be approximately one. For example, if BETA(1) is expected to lie between -10E10 and 10E10 then SCLB(1) should be set to 10E-10, while if BETA(1) is expected to lie between -10E-2 and -10E-4 then SCLB(1) should be set to 10E3. Except as noted in the next paragraph, the scale values specified for each BETA must be greater than zero; values less than or equal to zero will be treated as an input error.

By default, the scale values will be set using the algorithm given in section IX.A. The default values are invoked when SCLB(1) is set to any nonpositive value. If SCLB(1) > 0 then SCLB must contain a vector of NP values each greater than zero and the scale of BETA(K) is set to SCLB(K) for K=1,...,NP.

S 16. WD [<real> WD(LDWD,M)]

The doubly subscripted array that contains the values that specify the DELTA weights, D, which indicate how the DELTAs and EPSILONs of the observation (X,Y) are to be weighted in the weighted orthogonal distance, R (see eq.2). For example, WD(I,J) might be the the ratio of the precision of the Y(I) observation to that of the X(I,J) observation.

All elements of WD must be nonzero.

If WD(1,1) < zero then

only the first element of WD is ever referenced (in this case, WD can be a scalar) and

```
   D(I,J) = ABS(WD(1,1)) for I=1,...,N & J=1,...,M,
```

i.e., D(I,J) is constant and every DELTA is weighted equally with respect to each of the EPSILONs. When ABS(WD(1,1)) = 1, the DELTAs and EPSILONs are both weighted equally, possibly indicating the X and Y observations are equally precise.

If WD(1,1) > zero then

then all elements of WD must be greater than zero and the way WD is used to specify D depends on the value of the leading dimension of WD, i.e., on LDWD.

If LDWD = 1 then

WD must contain a 1 by M matrix of values, where for J=1,...,M

```
   D(I,J) = WD(1,J), I=1,...,N,
```

i.e., each column of D is constant. In this case, all elements of a given column of DELTA are weighted equally with respect to EPSILON, possibly reflecting that each observation within a given column of X is equally precise, but that the precision between columns varies.

If LDWD >= N then

WD must contain an N by M matrix of values, where

```
   D(I,J) = WD(I,J) for I=1,...,N & J=1,...,M,
```

i.e., each element of D is individually specified, possibly indicating that the individual observations of X vary significantly in precision both from each other and from the corresponding observations of Y.

17. LDWD [INTEGER LDWD]

The leading dimension of array WD.

LDWD must exactly equal one or must equal or exceed N; values of LDWD less than one or between two and N-1, inclusive, will be treated as an input error.

See subroutine argument WD for further details.

CD 18. W [<real> W(N)]

The singly subscripted array that contains the values that specify the observation error weights that can be used to compensate for unequal precision in the observation errors (see eq.3).

By default, the observation errors are unweighted, i.e., all of the weights are assumed to be identically equal to one. The default value is invoked when W(1) is set to any negative value. Other options for specifying W are described below.

If W(1) >= zero then

W must contain a vector of N values, where all elements of W must be greater than or equal to zero, and W(I), I=1,...,N, specifies the weight for the observation error R(I). Zero weights eliminate the corresponding observation from the analysis.

If W(1) < zero then

the default option is invoked, i.e., the observation errors are unweighted. In this case, only the first element of W is ever referenced and W can be a scalar.

D 19. JOB [INTEGER JOB]

The value used to specify problem initialization and computational methods. The user has the option of specifying five different aspects of the problem specification:

- whether the fit is to be by orthogonal distance regression (ODR) or by ordinary least squares (OLS);
- whether the user has supplied subroutine JAC to compute the necessary Jacobian matrices and whether the user supplied Jacobian matrices should be checked;
- whether the covariance matrix should be computed for the estimators of BETA;
- whether the DELTAs have been initialized by the user; and
- whether the fit is a restart.

By default:

- the solution will be found by ODR;
- the derivatives will be computed by finite differences;
- the covariance will be computed;
- the DELTAs will be initialized to zero; and
- the fit will not be a restart.

The default value is invoked by setting JOB to any value less than zero.

Setting JOB = 1 will have the same consequence as JOB = -1 except that the solution will be found by OLS.

If JOB > 0 then

JOB is assumed to be a 5 digit INTEGER with decimal expansion ABCDE, where each digit controls a different aspect of the problem specification.

Digit A indicates whether the fit is a restart.

A = 0 indicates fit is not a restart.

A > 0 indicates fit is a restart. The computations will continue from where they left off for another 10 iterations. If the fit is a restart then the elements of vector WORK must be exactly as returned from a previous call to ODRPACK. No error checking will be performed to verify this.

Digit B indicates whether the DELTAs have been initialized by the user.

B = 0 indicates DELTAs have not been initialized by user. The DELTAs will be initialized to zero.

B > 0 indicates DELTAs have been initialized by user. (See subroutine argument WORK.)

Digit C indicates whether the the covariance matrix of the estimators of the parameters BETA should be computed.

C = 0 indicates that the covariance matrix should be computed. (See subroutine argument IPRINT and section X.B.)

C > 0 indicates that the covariance matrix should not be computed.

Digit D indicates whether the user has supplied subroutine JAC to compute the necessary Jacobian matrices and whether the user supplied Jacobian matrices should be checked.

D = 0 indicates that the Jacobian matrices are to be computed by finite differences and that subroutine JAC will not be used.

D > 0 indicates that the user has supplied subroutine JAC to compute the necessary Jacobian matrices (see subroutine argument JAC).

If D = 1 the results of the user supplied routine will be checked for correctness. (Derivative checking requires one evaluation of user supplied subroutine JAC and at least NP+M evaluations of user supplied subroutine FUN.) Users who turn off the printed error reports by setting IPRINT=0 or LUNERR=0 should examine the information returned in IWORK to determine the results of the derivative checking procedure. (See subroutine argument INFO and section X.B.)

If D > 1 the results of the user supplied routine will not be checked for correctness.

Digit E indicates whether the fit is to be by orthogonal distance regression (ODR) or by ordinary least squares (OLS).

E = 0 indicates an ODR fit.

E > 0 indicates an OLS fit.

If JOB < 0 then

the "default" value will be used.

CD 20. NDIGIT [INTEGER NDIGIT]

The number of reliable decimal digits in the predicted values (F) computed by the user's model function. (See [Gill et al., 1981].)

By default, the value for NDIGIT is experimentally determined by ODRPACK using the first row of the user's data set that does not contain a zero observation. The computation of NDIGIT requires 5 evaluations of user supplied subroutine FUN. The default value is invoked when NDIGIT is set to any value outside the range [2, DIGITS], where DIGITS is the number of decimal digits carried by the user's computer for a single precision value when the SODR or SODRC are being used, and is the number carried for a double precision value when DODR or DODRC are being used.

CD 21. TAUFAC [<real> TAUFAC]

The value used to specify the initializing factor for the trust region radius. The trust region is the region in which the local approximation to the user's function is considered to be reliable. The diameter of this region is adaptively chosen at each iteration based on information from the previous iteration. At the first iteration, the initial diameter is set to the initializing factor times the length of the full Gauss-Newton step at the initial estimates.

By default, the initialization factor for the trust region radius is one, thus allowing the full Gauss-Newton step to be taken at the first iteration if it does, in fact, reduce the weighted sum of squares. The default value is invoked when TAUFAC is set to any value less than or equal to zero.

A value of TAUFAC greater than zero but less than one may be appropriate if, at the first iteration, the computed results overflow, or the function parameters, BETA, leave the region of interest in parameter space. Values of TAUFAC greater than one have the same effect on the computations as a value of one.

CD 22. SSTOL [<real> SSTOL]

The value used to specify the stopping tolerance for the convergence test based on relative change in the weighted sum of the squared observation errors (eq.3).

The "default" sum of squares convergence stopping tolerance is the square root of machine precision, where machine precision is defined as the smallest value e such that 1+e>1 on the computer being used. The default value is invoked when the user supplied value for SSTOL is outside the interval [e,1).

CD 23. PARTOL [<real> PARTOL]

The value used to specify the stopping tolerance for the convergence test based on relative change in the estimated parameters BETA and DELTA.

By default, the stopping tolerance for parameter convergence is (machine precision)**(2/3), where machine precision is defined as the smallest value e such that 1+e>1 on the computer being used.

The default value is invoked when the user supplied value for PARTOL is outside the interval [e,1).

CD 24. MAXIT [INTEGER MAXIT]

The value used to specify the maximum number of iterations allowed.

By default, the maximum number of iterations is 50. The default value is invoked when the user supplied value for MAXIT is less than or equal to zero.

D 25. IPRINT [INTEGER IPRINT]

The value used to control the generated computation reports, which are divided into three sections:
- the initial summary
- the iteration summary and
- the final summary.

The choice of content for each of these sections is described below.

By default, the computation reports include
- a "long" initial summary
- no iteration summary and
- a "short" final summary

The default value is invoked when the user supplied value for IPRINT is less than zero.

If IPRINT > 0 then

IPRINT is assumed to be a 4 digit INTEGER with decimal expansion ABCD, where each digit controls a different part of the generated reports.

Digit A indicates whether the initial summary will be generated.

A = 0 indicates the initial summary will not be generated.

A = 1 indicates a "short" initial summary will be generated that will include

* the values N, M and NP, the number of observations with nonzero weights, and the number of BETAs actually being estimated.

* the control values JOB, NDIGIT, TAUFAC, SSTOL, PARTOL, and MAXIT.

* the weighted sum of the squared observation errors, the sum of the squared weighted DELTAs and the sum of the squared weighted EPSILONs at the initial values of BETA and DELTA.

A > 1 indicates a "long" initial summary will be generated, which includes all the information found in the "short" initial summary and, in addition, includes

* a summary of the independent variable data, organized by column.

* the first and last observation of the dependent variable and the first and last observation error weight.

* for each function parameter BETA, the initial value, whether or not the parameter is treated as fixed or not, and the scale value to be used.

Digit B indicates whether the iteration summary will be generated.

B = 0 indicates no iteration summary will be generated.

B = 1 indicates a "short" 1 line, 68 column iteration summary will be generated every Cth iteration beginning with iteration one. This summary will list

* the number of function evaluations.

* the weighted sum of the squared observation errors at the current point.

* the actual relative reduction in the weighted sum of the squared observation errors due to the most recently tried step (used to check for sum of squares convergence).

* the predicted relative reduction in the weighted sum of the squared observation errors due to the most recently tried step (used to check for sum of squares convergence).

* the ratio of the trust region radius to the norm of the BETAs and DELTAs, which is an upper bound on the relative change in the estimated values possible at the next step (used to check for parameter convergence).

* whether the step was a Gauss-Newton step.

B > 1 indicates an [NP/3] line, 125 column iteration summary will be generated every Cth iteration beginning with iteration 1. This summary lists all of the information found in the "short" iteration summary and, in addition, includes

* current values of the BETAs. (Note that, at the last iteration, the values listed for BETA will be those that produced the actual and predicted relative reductions shown only if the most recently tried step did in fact make the fit better. If not, then the values of BETA are those that produced the best fit.

Digit C indicates the frequency of the iteration summary.

C = 0 indicates no iteration summary will be generated, even if the value of digit B is nonzero.

C > 0 indicates an iteration summary will be generated every Cth iteration beginning with iteration one.

Digit D indicates whether the final summary will be generated.

D = 0 indicates the final summary will not be generated.

D = 1 indicates a "short" final summary will be generated, which includes

* the stopping condition.

* the number of iterations, the number of function evaluations and, if the Jacobian was supplied by the user, the number of Jacobian evaluations at the time the computations stopped.

* the condition number of the problem at the time the computations stopped.

* the rank deficiency of the model at the time the computations stopped.

* the final weighted sum of the squared observation errors, the final sum of the squared weighted DELTAs, the final sum of the squared weighted EPSILONs, and if the covariance matrix was computed, the estimated residual variance of the fit, RVAR, and the associated degrees of freedom, DF, where

```
            1     N
RVAR = -- * SUM (W(I)*R(I))**2
            DF    I=1
```

DF = the number of observations with nonzero weighted derivatives with respect to either BETA or DELTA minus the number of parameters actually estimated.

* the final values of BETA, and, if the covariance matrix was computed, the standard errors for the estimators of BETA. (See subroutine argument JOB.) The standard errors are computed as the square root of the diagonal elements of the variance covariance matrix VCV,

```
   VCV = RVAR * inv( trans(FJACB)*OMEGA*FJACB )
```

where

RVAR is defined above;

FJACB is the derivative of FN(X(I,J)+DELTA(I,J);BETA) with respect to BETA, evalutated at the solution;

OMEGA is the diagonal matrix which has (I,I)th element

$$\mathrm{OMEGA}(I,I) = \frac{W(I)^{2}}{1 + \sum_{J=1}^{M} \frac{\mathrm{FJACX}(I,J)^{2}}{D(I,J)^{2}}}$$

with FJACX(I,J) the derivative of FN(X(I,J)+DELTA(I,J);BETA) with respect to DELTA(I,J), evaluated at the solution (for ordinary least squares, OMEGA(I,I) reduces to W(I)**2);

inv(.) indicates the inverse of the designated matrix; and

trans(.) indicates the transpose of the designated matrix.

Note that the covariance matrix is an approximation based on a linearization of the model in the neighborhood of the solution. The validity of the approximation depends on the nonlinearity of the model, the variance and distribution of the errors, and the data itself. Confidence regions and intervals computed using the covariance matrix are often acceptable, but can be very inaccurate in some cases. When reliable confidence intervals and regions are required, other more accurate, but more computationally expensive methods of constructing them should be used. (See, e.g., Boggs and Donaldson [1989], Donaldson and Schnabel [1987], Efron [1985], and Fuller [1987].)

* the first 32 values of EPSILON, and the first 32 values of each column of DELTA.

D > 1 indicates a "long" final summary will be generated, which includes the same information as the "short" final summary except that

* the values of all of the EPSILONs and DELTAs are listed.

If IPRINT < 0 then

the default reports will be generated.

If IPRINT = 0 then

the no reports will be generated.

D 26. LUNERR [INTEGER LUNERR]

The value used to specify the logical unit number of the file to be used for error messages.

By default, the error messages are generated on unit 6. The default value is invoked when the user supplied value for LUNERR is less than zero.

If LUNERR > 0   the error messages will be generated on unit LUNERR.

If LUNERR = 0   no error messages will be generated.

If LUNERR < 0   the "default" unit number will be used.

D 27. LUNRPT [INTEGER LUNRPT]

The value used to specify the logical unit number of the file to be used for computation reports.

By default, the computation reports are generated on unit 6. The default value is invoked when the user supplied value for LUNRPT is less than zero.

If LUNRPT > 0   the computation reports will be generated on unit LUNRPT.

If LUNRPT = 0   no computation reports will be generated.

If LUNRPT < 0   the "default" unit number will be used.

28. WORK [<real> WORK(LWORK)]

The singly subscripted array used for <real> work space and the array in which various computed values are returned. The smallest acceptable dimension of WORK is given below in the definition of subroutine argument LWORK.

The work area does not need to be initialized by the user unless the user wishes to initialize the DELTAs.

The first N*M locations of WORK contain the values for DELTA. An easy way to access these values, either for initialization (as indicated by digit B of subroutine argument JOB) or for analysis upon return from ODRPACK, is to include in the user's program the declaration statements

```
<real> DELTA(<N>,<M>)
EQUIVALENCE (WORK(1), DELTA(1,1))
```

where <N> indicates the first dimension of the array DELTA must be exactly the number of observations, N; and
<M> indicates the second dimension of the array DELTA must be exactly the number of columns, M, of the independent variable, X.

This allows the error associated with observation X(I,J) of the independent variable matrix to be accessed as DELTA(I,J) rather than as WORK(I+(J-1)*N). The input values of array DELTA will be over written by the final estimates of the errors in the independent variable matrix when this equivalencing method is used.

Other values returned in array WORK may also be of general interest and can be accessed as described below in section X.

N.B., if the fit is a restart, i.e., if digit A of subroutine argument JOB is nonzero, then all elements of vector WORK, including the values of DELTA, must be exactly as returned from a previous call to ODRPACK.

29. LWORK [INTEGER LWORK]

The length of array WORK. LWORK must equal or exceed

17 + 7*N + 10*N*M + 2*N*NP + 8*NP .

Values of LWORK less than this value will be treated as an input error.

30. IWORK [INTEGER IWORK(LIWORK)]

The singly subscripted array used for INTEGER work space and the array in which various computed values are returned. The smallest acceptable dimension of IWORK is given below in the definition of subroutine argument LIWORK.

Certain values returned in array IWORK are of general interest and can be accessed as described below in section X. In particular, the results of the derivative checking procedure are encoded in IWORK, and may be useful if ODRPACK's error reports have been suppressed.

31. LIWORK [INTEGER LIWORK]

The length of array IWORK. LIWORK must equal or exceed

19 + 2*NP + M .

Values of LIWORK less than this value will be treated as an input error.

32. INFO [INTEGER INFO]

The argument used to indicate why the computations stopped.

If 1 <= INFO <= 3 then

the program converged satisfactorily. The convergence condition met is indicated by the value of INFO as follows.
INFO = 1 : sum of squares convergence
INFO = 2 : parameter convergence
INFO = 3 : sum of squares convergence and parameter convergence

If INFO = 4 then

the program reached the maximum number of iterations allowed without meeting one of the convergence conditions.

If INFO > 4 and INFO < 10000 then

the results from ODRPACK are questionable. In this case, INFO is a 4 digit INTEGER with decimal expansion ABCD, where digit D indicates the actual stopping condition, and the nonzero values of digits A, B and C indicate what questionable conditions were found.

Digit A = 1 indicates

the ODRPACK Jacobian matrix checking procedure determined that the correctness of the user supplied Jacobian matrices is questionable. This occurs when the derivative is exactly zero or when the numerical derivative used in the checking procedure is believed to be inaccurate. (Zero valued derivatives are questionable because they could indicate that the initial values of the function parameters BETA might be hiding an error in the derivative, such as could occur if the initial value of one of the parameters were zero.) Users should examine the ODRPACK error reports or the encoded values of IWORK (see section X.B) to determine the cause of the questionable results, and then examine subroutine JAC to insure that there is not an error in the user supplied derivatives that could be adversely affecting the least squares results.

Digit B = 1 indicates

the the most recently tried values of BETA and/or X+DELTA were unacceptable, as indicated by the returned value of ISTOPF from user supplied subroutine FUN (see argument FUN).

Digit C > 0 indicates

the Jacobian with respect to the function parameters BETA is not full rank at the solution.
If C=1 the rank is greater than zero but less than the number of parameters being estimated.
If C=2 the rank is zero, indicating that the results of user supplied subroutines FUN and/or JAC are unaffected by changes in the unfixed function parameters (BETA), and therefore indicating that there is a probable error in these user supplied subroutines.

Digit D > 0 indicates

the actual stopping condition.
If D=1 the sum of squares convergence criteria was met.
If D=2 the parameter convergence criteria was met.
If D=3 the sum of squares convergence criteria and the parameter convergence criteria were met.
If D=4 the program reached the maximum number of iterations allowed without meeting one of the convergence conditions.

If INFO > 9999 then

fatal errors were detected that required that the computations be stopped. In this case, INFO is a 5 digit INTEGER with decimal expansion ABCDE, where each nonzero digit indicates a different error condition.

Digit A = 1 indicates an error was detected in the arguments used to specify the problem size.

When digit A = 1 then

- digit B = 1 indicates N < 1
- digit C = 1 indicates M < 1
- digit D = 1 indicates NP < 1 or NP > N

Digit A = 2 indicates an error was detected in the arguments used to specify array dimensions.

When digit A = 2 then

- digit B = 1 indicates LDX < N
- digit C > 0 indicates LDIFX, LDSCLD and/or LDWD are unacceptable (see definitions of LDIFX, LDSCLD and LDWD for acceptable values), where
  - if C=1 LDIFX is bad
  - if C=2 LDSCLD is bad
  - if C=3 LDIFX & LDSCLD are bad
  - if C=4 LDWD is bad
  - if C=5 LDIFX & LDWD are bad
  - if C=6 LDSCLD & LDWD are bad
  - if C=7 LDIFX, LDSCLD & LDWD are bad
- digit D = 1 indicates LWORK is too small (see definition of LWORK for smallest acceptable value)
- digit E = 1 indicates LIWORK is too small (see definition of LIWORK for smallest acceptable value)

Digit A = 3 indicates an error was detected in the arguments used to specify scaling and/or the in the arguments used to specify the weights.

When digit A = 3 then

- digit B = 1 indicates an error in SCLD (see definition of SCLD for reasonable values)
- digit C = 1 indicates an error in SCLB (see definition of SCLB for reasonable values)
- digit D > 1 indicates an error in W, where
  - if D=1 one or more of the elements of W are invalid (see definition of W for reasonable values)
  - if D=2 the number of nonzero values in W is less than NP
- digit E = 1 indicates an error in WD (see definition of WD for reasonable values)

Digit A = 4 indicates an error was detected in the user supplied Jacobian matrices.

When digit A = 4 then

- digit B = 1 indicates an error in the Jacobian matrix with respect to BETA (see the generated error reports, or section X.B for locations in IWORK that indicate which derivatives are in error)

- digit C = 1 indicates an error in the Jacobian matrix with respect to X (see the generated error reports, or section X.B for locations in IWORK that indicate which derivatives are in error)

Digit A = 5 indicates the values of BETA and/or X+DELTA were identified as unacceptable by user supplied subroutine FUN or JAC.

When digit A = 5 then

- digit B > 0 indicates the computations were stopped in user supplied subroutine FUN, where
  - if B=1 variable ISTOPF was returned with a negative value from subroutine FUN when it was invoked during the regression procedure, indicating that the user wanted the computations stopped
  - if B=2 variable ISTOPF was returned with a nonzero value when subroutine FUN was invoked using the initial estimates of BETA and DELTA supplied by the user, so no further computations could be performed
  - if B=3 variable ISTOPF was returned with a nonzero value when subroutine FUN was was invoked during the computation of the number of reliable digits in the predicted values (F) returned from subroutine FUN, indicating that changes in the initial estimates of BETA(K), K=1,NP, as small as 2*BETA(K)*sqrt(e), where e is defined as the smallest value such that 1+e>1 on the computer being used, prevent subroutine FUN from being properly evaluated
  - if B=4 variable ISTOPF was returned with a nonzero value when subroutine FUN was was invoked during the derivative checking procedure, indicating that changes in the initial estimates of BETA(K), K=1,NP, small as max[BETA(K),1/SCLB(K)]*10**(-NETA/2), and/or of DELTA(I,J), i=1,N and j=1,M, as small as max[DELTA(I,J), 1/SCLD(I,J)]*10**(-NETA/2), where NETA is defined to be the number of reliable digits in predicted values (F) returned from subroutine FUN, prevent subroutine fun from being properly evaluated

digit C > 0 indicates the computations were stopped in user supplied subroutine JAC, where

if C=1 variable ISTOPJ was returned with a nonzero value from subroutine JAC when it was invoked during the regression procedure, indicating that the user wanted the computations stopped

if C=2 variable ISTOPJ was returned with a nonzero value from subroutine JAC when it was invoked using the initial estimates of BETA and DELTA supplied by the user, so no further computations could be performed

VIII. EXAMPLES

The following sample programs use DODR and DODRC to solve exercise I on page 521 and 522 of Draper and Smith [1981]. The program calling DODR uses the default option of computing the derivatives by finite differences, while the program calling DODRC uses analytic derivatives. Note that the results of these two examples are not identical, primarily because the DODRC example has "fixed" one column of the independent variable. Finite difference derivatives generally cause very little change in the results from those obtained using analytic derivatives.

Users are encouraged to extract these examples from the online ODRPACK documentation, and to then modify them as necessary to form their own ODRPACK drivers. (Single precision sample programs can be easily generated from these two programs by changing all DOUBLE PRECISION variables to REAL, and substituting SODR for DODR and SODRC for DODRC.) Note especially that by using parameters MAXN, MAXM and MAXNP to specify the largest problem the program can solve without modification, and by specifying LWORK and LIWORK exactly as shown, the user greatly reduces the number of changes that must be made to the program in order to solve a larger problem.

VIII.A DODR Example Program, Data and ODRPACK Generated Report

User supplied code for DODR example:

```
      PROGRAM SAMPLE

C  SET PARAMETERS FOR MAXIMUM PROBLEM SIZE HANDLED BY THIS DRIVER
C  WHERE  MAXN IS THE MAXIMUM NUMBER OF OBSERVATIONS ALLOWED
C         MAXM IS THE MAXIMUM NUMBER OF COLUMNS IN THE
C              INDEPENDENT VARIABLE ALLOWED
C        MAXNP IS THE MAXIMUM NUMBER OF FUNCTION PARAMETERS
C              ALLOWED
C          LDX IS THE LEADING DIMENSION OF ARRAY X
C         LDWD IS THE LEADING DIMENSION OF ARRAY WD
C        LWORK IS THE DIMENSION OF VECTOR WORK
C       LIWORK IS THE DIMENSION OF VECTOR IWORK

C...PARAMETERS
      INTEGER
     +   MAXN,MAXM,MAXNP,LDX,LDWD,LWORK,LIWORK
      PARAMETER
     +   (MAXN=15,
     +   MAXM=5,
     +   MAXNP=5,
     +   LDX=MAXN,
     +   LDWD=1,
     +   LWORK = 17 + 7*MAXN + 10*MAXN*MAXM + 2*MAXN*MAXNP + 8*MAXNP,
     +   LIWORK = 19 + 2*MAXNP + MAXM)

C  DECLARE USER-SUPPLIED SUBROUTINES AND
C  ALL OTHER NECESSARY VARIABLES AND ARRAYS

C...LOCAL SCALARS
      INTEGER
     +   I,INFO,IPRINT,J,JOB,LUNERR,LUNRPT,M,N,NP

C...LOCAL ARRAYS
      DOUBLE PRECISION
     +   BETA(MAXNP),WD(LDWD,MAXM),WORK(LWORK),
     +   X(LDX,MAXM),Y(LDX)
      INTEGER
     +   IWORK(LIWORK)

C...EXTERNAL SUBROUTINES
      EXTERNAL
     +   DODR,FUN,JAC


      OPEN(UNIT=5,FILE='DATA1')
      OPEN(UNIT=6,FILE='REPORT')

C  READ NUMBER OF OBSERVATIONS
C       NUMBER OF COLUMNS OF DATA IN THE INDEPENDENT VARIABLE
C       NUMBER OF PARAMETERS
C       OBSERVED VALUES OF INDEPENDENT AND DEPENDENT VARIABLES
C       STARTING VALUES OF FUNCTION PARAMETERS

      READ (5,*) N,M,NP
      READ (5,*) ((X(I,J),I=1,N),J=1,M)
      READ (5,*) (Y(I),I=1,N)
      READ (5,*) (BETA(I),I=1,NP)

C  SPECIFY DELTA WEIGHTS

      WD(1,1) = 3.0D0
      WD(1,2) = 5.0D0

C  SET CONTROL VALUES TO INVOKE DEFAULT SETTING

      JOB = -1
      IPRINT = -1
      LUNERR = -1
      LUNRPT = -1

C  COMPUTE ODR SOLUTION USING FINITE-DIFFERENCE DERIVATIVES

      CALL DODR
     +   (FUN,JAC,
     +   N,M,NP,
     +   X,LDX,
     +   Y,
     +   BETA,
     +   WD,LDWD,
     +   JOB,
     +   IPRINT,LUNERR,LUNRPT,
     +   WORK,LWORK,IWORK,LIWORK,
     +   INFO)

      END
      SUBROUTINE FUN(N,NP,M,BETA,XPLUSD,LDXPD,F,ISTOPF)

C  INPUT ARGUMENTS
C  (WHICH MUST NOT BE CHANGED BY THIS ROUTINE)
C     INTEGER N,NP,M,LDXPD
C     DOUBLE PRECISION BETA(NP),XPLUSD(LDXPD,M)
C  OUTPUT ARGUMENTS
C     DOUBLE PRECISION F(N)
C     INTEGER ISTOPF

C...SCALAR ARGUMENTS
      INTEGER
     +   ISTOPF,LDXPD,M,N,NP

C...ARRAY ARGUMENTS
      DOUBLE PRECISION
     +   BETA(NP),F(N),XPLUSD(LDXPD,M)

C...LOCAL SCALARS
      INTEGER
     +   I

C...INTRINSIC FUNCTIONS
      INTRINSIC
     +   EXP


      DO 10 I = 1, N
         IF (XPLUSD(I,2).NE.0.0D0) THEN
            F(I) = EXP(-BETA(1)*XPLUSD(I,1)*
     +             EXP(-BETA(2)*
  +                  (1.0D0/XPLUSD(I,2) - 1.0D0/620.0D0)))
      ELSE
         ISTOPF = 1
         RETURN
      END IF
10 CONTINUE
   ISTOPF = 0

   RETURN
   END
```

User supplied data (file DATA1):

```
    8    2    2
  109.0   65.0 1180.0   66.0
 1270.0   69.0 1230.0   68.0
  600.0  640.0  600.0  640.0
  600.0  640.0  600.0  640.0
  0.912  0.382  0.397  0.376
  0.342  0.358  0.348  0.376
0.01155 5000.0
```

Report generated by DODR example program, using a Sun 3 Workstation:

```
*******************************************************
* ODRPACK VERSION 1.71 OF 07-27-89 (DOUBLE PRECISION) *
*******************************************************


INITIAL SUMMARY FOR FIT BY METHOD OF ODR
========================================


PROBLEM SIZE:
-------------

NUMBER OF OBSERVATIONS                                8
NUMBER OF OBSERVATIONS WITH NONZERO WEIGHTS           8
NUMBER OF COLUMNS OF DATA IN INDEPENDENT VARIABLE     2
NUMBER OF FUNCTION PARAMETERS                         2
NUMBER OF UNFIXED FUNCTION PARAMETERS                 2


INDEPENDENT VARIABLE AND DELTA WEIGHT SUMMARY:
----------------------------------------------

                              COLUMN   1                COLUMN   2
                          OBS 1        OBS N        OBS 1        OBS N
              X -   0.10900D+03  0.68000D+02  0.60000D+03  0.64000D+03
          FIXED -            NO           NO           NO           NO
  INITIAL DELTA -   0.00000D+00  0.00000D+00  0.00000D+00  0.00000D+00
    DELTA SCALE -   0.91743D-02  0.14706D-01  0.15625D-02  0.15625D-02
  DELTA WEIGHTS -   0.30000D+01  0.30000D+01  0.50000D+01  0.50000D+01


DEPENDENT VARIABLE AND OBSERVATIONAL ERROR WEIGHT SUMMARY:
----------------------------------------------------------

                          OBS 1        OBS N
              Y -   0.91200D+00  0.37600D+00
OBS. ERROR WTS. -   0.10000D+01  0.10000D+01


FUNCTION PARAMETER SUMMARY:
---------------------------

       INDEX -                1               2
INITIAL BETA -   0.11550000D-01  0.50000000D+04
       FIXED -               NO              NO
  BETA SCALE -   0.86580087D+02  0.20000000D-03
```

```
CONTROL VALUES AND STOPPING CRITERIA:
--------------------------------------


      *
   JOB    NDIGIT    TAUFAC     SSTOL    PARTOL  MAXIT
 00000        15  0.10D+01  0.15D-07  0.37D-10     50

*
 A.  FIT IS NOT A RESTART.
 B.  DELTAS ARE INITIALIZED TO ZERO.
 C.  THE COVARIANCE MATRIX OF THE PARAMETER ESTIMATORS
     WILL BE COMPUTED AT THE SOLUTION.
 D.  DERIVATIVES ARE COMPUTED BY FINITE DIFFERENCES.
 E.  FIT IS BY METHOD OF ORTHOGONAL DISTANCE REGRESSION.


INITIAL SUMS OF SQUARES:
------------------------

SUM OF SQUARED WEIGHTED OBSERVATIONAL ERRORS    0.67662011D+00
SUM OF SQUARED WEIGHTED DELTAS                  0.00000000D+00
SUM OF SQUARED WEIGHTED EPSILONS                0.67662011D+00





FINAL SUMMARY FOR FIT BY METHOD OF ODR
======================================



STOPPING CONDITION (INFO =      1):
-----------------------------------

THE RELATIVE CHANGE IN THE SUM OF THE SQUARED
WEIGHTED OBSERVATIONAL ERRORS IS LESS THAN SSTOL

                            CONDITION
      NUMBER OF  NUMBER OF     NUMBER        RANK
     ITERATIONS   FN EVALS  (INVERSE)  DEFICIENCY
              5         42 0.1888D-06           0


FINAL SUMS OF SQUARES:
----------------------

SUM OF SQUARED WEIGHTED OBSERVATIONAL ERRORS    0.75382323D-03
SUM OF SQUARED WEIGHTED DELTAS                  0.23542098D-07
SUM OF SQUARED WEIGHTED EPSILONS                0.75379969D-03

ESTIMATED RESIDUAL VARIANCE                     0.12563720D-03
(    6 DEGREES OF FREEDOM)
```

```
ESTIMATED BETA(J), J = 1, ..., NP:
----------------------------------

            J           BETA(J)     STD. DEV. BETA(J)
            1    0.36579727D-02        0.42219552D-04
            2    0.27627327D+05        0.22245631D+03


ESTIMATED EPSILON(I) AND DELTA(I,*), I = 1, ..., N:
--------------------------------------------------

    I      EPSILON(I)       DELTA(I,1)       DELTA(I,2)
    1  0.16752445D-02  0.14086172D-06  0.42418826D-06
    2  0.20434718D-02  0.12838222D-05  0.20262810D-05
    3 -0.20690085D-01 -0.71652290D-06 -0.23358824D-04
    4  0.24305832D-02  0.15047092D-05  0.24114481D-05
    5  0.72777482D-02  0.23393281D-06  0.82079307D-05
    6  0.40793264D-02  0.24162846D-05  0.40483550D-05
    7  0.13043071D-01  0.43337343D-06  0.14726726D-04
    8 -0.85499649D-02 -0.51394679D-05 -0.84861061D-05
```

VIII.B DODRC Example Program, Data and ODRPACK Generated Report

User supplied code for DODRC example:

```
      PROGRAM SAMPLE

C   SET PARAMETERS FOR MAXIMUM PROBLEM SIZE HANDLED BY THIS DRIVER
C   WHERE  MAXN IS THE MAXIMUM NUMBER OF OBSERVATIONS ALLOWED
C          MAXM IS THE MAXIMUM NUMBER OF COLUMNS IN THE
C               INDEPENDENT VARIABLE ALLOWED
C         MAXNP IS THE MAXIMUM NUMBER OF FUNCTION PARAMETERS
C               ALLOWED
C           LDX IS THE LEADING DIMENSION OF ARRAY X
C        LDSCLD IS THE LEADING DIMENSION OF ARRAY SCLD
C          LDWD IS THE LEADING DIMENSION OF ARRAY WD
C         LDIFX IS THE LEADING DIMENSION OF ARRAY IFIXX
C         LWORK IS THE DIMENSION OF VECTOR WORK
C        LIWORK IS THE DIMENSION OF VECTOR IWORK

C...PARAMETERS
      INTEGER
     +   MAXN,MAXM,MAXNP,LDSCLD,LDIFX,LDWD,LWORK,LIWORK
      PARAMETER
     +   (MAXN=15,
     +   MAXM=5,
     +   MAXNP=5,
     +   LDSCLD=1,
     +   LDWD=1,
     +   LDIFX=1,
     +   LWORK=17 + 7*MAXN + 10*MAXN*MAXM + 2*MAXN*MAXNP + 8*MAXNP,
     +   LIWORK=19 + 2*MAXNP + MAXM)

C   DECLARE USER-SUPPLIED SUBROUTINES AND
C   ALL OTHER NECESSARY VARIABLES AND ARRAYS

C...LOCAL SCALARS
      DOUBLE PRECISION
     +   PARTOL,SSTOL,TAUFAC
      INTEGER
     +   I,INFO,IPRINT,J,JOB,LDX,LUNERR,LUNRPT,M,MAXIT,N,NDIGIT,NP

C...LOCAL ARRAYS
      DOUBLE PRECISION
     +   BETA(MAXNP),WD(LDWD,MAXM),SCLB(MAXNP),
     +   SCLD(LDSCLD,MAXM),W(MAXN),WORK(LWORK),X(MAXN,MAXM),Y(MAXN)
      INTEGER
     +   IFIXB(MAXNP),IFIXX(LDIFX,MAXM),IWORK(LIWORK)

C...EXTERNAL SUBROUTINES
      EXTERNAL
     +   DODRC,FUN,JAC


      OPEN(UNIT=5,FILE='DATA1')
      OPEN(UNIT=6,FILE='REPORT')

C   SPECIFY LEADING DIMENSION OF ARRAY X

      LDX = MAXN
```

```
C  READ NUMBER OF OBSERVATIONS
C       NUMBER OF COLUMNS OF DATA IN THE INDEPENDENT VARIABLE
C       NUMBER OF PARAMETERS
C       OBSERVED VALUES OF INDEPENDENT AND DEPENDENT VARIABLES
C       STARTING VALUES OF FUNCTION PARAMETERS

      READ (5,*) N,M,NP
      READ (5,*) ((X(I,J),I=1,N),J=1,M)
      READ (5,*) (Y(I),I=1,N)
      READ (5,*) (BETA(I),I=1,NP)

C  FIX SECOND COLUMN OF INDEPENDENT VARIABLE AT OBSERVED VALUES

      IFIXX(1,1) = 1
      IFIXX(1,2) = 0

C  SPECIFY USE OF DEFAULT SCALING

      SCLD(1,1) = -1.0D0
      SCLB(1) = -1.0D0

C  INDICATE ALL BETA'S ARE TO BE ESTIMATED

      IFIXB(1) = -1

C  SPECIFY WEIGHTS

      WD(1,1) = 3.0D0
      WD(1,2) = 5.0D0
      W(1) = -1.0D0

C  SET CONTROL VALUES AND STOPPING CRITERIA

      JOB = 10
      NDIGIT = -1
      TAUFAC = -1.0D0
      SSTOL = -1.0D0
      PARTOL = -1.0D0
      MAXIT = -1
      IPRINT = 1111
      LUNERR = -1
      LUNRPT = -1

C  COMPUTE ODR SOLUTION USING USER-SUPPLIED ANALYTIC DERIVATIVES

      CALL DODRC
     +   (FUN,JAC,
     +   N,M,NP,
     +   X,LDX,IFIXX,LDIFX,SCLD,LDSCLD,
     +   Y,
     +   BETA,IFIXB,SCLB,
     +   WD,LDWD,W,
     +   JOB,NDIGIT,TAUFAC,
     +   SSTOL,PARTOL,MAXIT,
     +   IPRINT,LUNERR,LUNRPT,
     +   WORK,LWORK,IWORK,LIWORK,
     +   INFO)

      END
      SUBROUTINE FUN(N,NP,M,BETA,XPLUSD,LDXPD,F,ISTOPF)
```

```
C  INPUT ARGUMENTS
C  (WHICH MUST NOT BE CHANGED BY THIS ROUTINE)

C     INTEGER N,NP,M,LDXPD
C     DOUBLE PRECISION BETA(NP),XPLUSD(LDXPD,M)

C  OUTPUT ARGUMENTS

C     DOUBLE PRECISION F(N)
C     INTEGER ISTOPF

C...SCALAR ARGUMENTS
      INTEGER
     +   ISTOPF,LDXPD,M,N,NP

C...ARRAY ARGUMENTS
      DOUBLE PRECISION
     +    BETA(NP),F(N),XPLUSD(LDXPD,M)

C...LOCAL SCALARS
      INTEGER
     +    I

C...INTRINSIC FUNCTIONS
      INTRINSIC
     +    EXP


      DO 10 I = 1, N
         IF (XPLUSD(I,2).NE.0.0D0) THEN
            F(I) = EXP(-BETA(1)*XPLUSD(I,1)*
     +             EXP(-BETA(2)*
     +                  (1.0D0/XPLUSD(I,2) - 1.0D0/620.0D0)))
         ELSE
            ISTOPF = 1
            RETURN
         END IF
   10 CONTINUE
      ISTOPF = 0

      RETURN
      END
      SUBROUTINE JAC(N,NP,M,BETA,XPLUSD,LDXPD,
     +               FJACB,LDFJB,ISODR,FJACX,LDFJX,ISTOPJ)

C  INPUT ARGUMENTS
C  (WHICH MUST NOT BE CHANGED BY THIS ROUTINE)

C     INTEGER N,NP,M,LDXPD
C     DOUBLE PRECISION BETA(NP),XPLUSD(LDXPD,M)
C     LOGICAL ISODR

C  OUTPUT ARGUMENTS

C     DOUBLE PRECISION FJACB(LDFJB,NP),FJACX(LDFJX,M)
C     INTEGER ISTOPJ

C...SCALAR ARGUMENTS
      INTEGER
     +   ISTOPJ,LDFJB,LDFJX,LDXPD,M,N,NP
      LOGICAL
     +   ISODR

C...ARRAY ARGUMENTS
      DOUBLE PRECISION
     +   BETA(NP),FJACB(LDFJB,NP),FJACX(LDFJX,M),XPLUSD(LDXPD,M)

C...LOCAL SCALARS
      DOUBLE PRECISION
     +   FAC1,FAC2,FAC3,FAC4
      INTEGER
     +   I

C...INTRINSIC FUNCTIONS
      INTRINSIC
     +   EXP


      DO 10 I=1,N
         FAC1 = 1.0D0/XPLUSD(I,2) - 1.0D0/620.0D0
         FAC2 = EXP(-BETA(2)*FAC1)
         FAC3 = BETA(1)*XPLUSD(I,1)
         FAC4 = EXP(-FAC3*FAC2)

         FJACB(I,1) = -FAC4*XPLUSD(I,1)*FAC2
         FJACB(I,2) = FAC4*FAC3*FAC2*FAC1

         IF (ISODR) THEN
            FJACX(I,1) = -FAC4*BETA(1)*FAC2
            FJACX(I,2) = -FAC4*FAC3*FAC2*BETA(2)/XPLUSD(I,2)**2
         END IF
   10 CONTINUE
      ISTOPJ = 0

      RETURN
      END
```

User supplied data (file DATA1):

```
    8    2    2
  109.0   65.0 1180.0   66.0
 1270.0   69.0 1230.0   68.0
  600.0  640.0  600.0  640.0
  600.0  640.0  600.0  640.0
  0.912  0.382  0.397  0.376
  0.342  0.358  0.348  0.376
0.01155 5000.0
```

Report generated by DODRC example program, using a Sun 3 Workstation:

```
*******************************************************
* ODRPACK VERSION 1.71 OF 07-27-89 (DOUBLE PRECISION) *
*******************************************************



INITIAL SUMMARY FOR FIT BY METHOD OF ODR
========================================


PROBLEM SIZE:
-------------

NUMBER OF OBSERVATIONS                                8
NUMBER OF OBSERVATIONS WITH NONZERO WEIGHTS           8
NUMBER OF COLUMNS OF DATA IN INDEPENDENT VARIABLE     2
NUMBER OF FUNCTION PARAMETERS                         2
NUMBER OF UNFIXED FUNCTION PARAMETERS                 2


CONTROL VALUES AND STOPPING CRITERIA:
--------------------------------------

      *
   JOB    NDIGIT    TAUFAC     SSTOL    PARTOL  MAXIT
 00010        15  0.10D+01  0.15D-07  0.37D-10     50

*
 A.  FIT IS NOT A RESTART.
 B.  DELTAS ARE INITIALIZED TO ZERO.
 C.  THE COVARIANCE MATRIX OF THE PARAMETER ESTIMATORS
     WILL BE COMPUTED AT THE SOLUTION.
 D.  DERIVATIVES ARE SUPPLIED BY USER.
     USER-SUPPLIED DERIVATIVES WERE CHECKED.
     THE DERIVATIVES APPEAR TO BE CORRECT.
 E.  FIT IS BY METHOD OF ORTHOGONAL DISTANCE REGRESSION.


INITIAL SUMS OF SQUARES:
------------------------

SUM OF SQUARED WEIGHTED OBSERVATIONAL ERRORS    0.67662011D+00
SUM OF SQUARED WEIGHTED DELTAS                  0.00000000D+00
SUM OF SQUARED WEIGHTED EPSILONS                0.67662011D+00
```

```
ITERATION REPORTS FOR FIT BY METHOD OF ODR
==========================================


        CUM.                 ACT. REL.   PRED. REL.
 IT.  NO. FN     WEIGHTED   SUM-OF-SQS   SUM-OF-SQS              G-N
NUM.   EVALS   SUM-OF-SQS    REDUCTION    REDUCTION  TAU/PNORM  STEP
----  ------  -----------  -----------  -----------  ---------  ----

   1      12  0.19694D+00   0.7089D+00   0.4162D+00  0.151D+01   YES
   2      13  0.18660D-02   0.9905D+00   0.9957D+00  0.671D+00   YES
   3      14  0.75385D-03   0.5960D+00   0.5961D+00  0.463D-01   YES
   4      15  0.75385D-03   0.3659D-06   0.3659D-06  0.224D-04   YES
   5      16  0.75385D-03   0.3715D-13   0.3892D-13  0.482D-08   YES




FINAL SUMMARY FOR FIT BY METHOD OF ODR
======================================


STOPPING CONDITION (INFO =      1):
-----------------------------------

THE RELATIVE CHANGE IN THE SUM OF THE SQUARED
WEIGHTED OBSERVATIONAL ERRORS IS LESS THAN SSTOL

                                      CONDITION
      NUMBER OF  NUMBER OF  NUMBER OF    NUMBER        RANK
     ITERATIONS   FN EVALS  JAC EVALS (INVERSE)  DEFICIENCY
              5         17          7 0.1888D-06          0


FINAL SUMS OF SQUARES:
----------------------

SUM OF SQUARED WEIGHTED OBSERVATIONAL ERRORS    0.75384644D-03
SUM OF SQUARED WEIGHTED DELTAS                  0.33248273D-09
SUM OF SQUARED WEIGHTED EPSILONS                0.75384611D-03

ESTIMATED RESIDUAL VARIANCE                     0.12564107D-03
(    6 DEGREES OF FREEDOM)


ESTIMATED BETA(J), J = 1, ..., NP:
----------------------------------

            J          BETA(J)     STD. DEV. BETA(J)
            1   0.36579727D-02        0.42219603D-04
            2   0.27627326D+05        0.22245657D+03
```

```
ESTIMATED EPSILON(I) AND DELTA(I,*), I = 1, ..., N:
---------------------------------------------------

    I      EPSILON(I)       DELTA(I,1)       DELTA(I,2)
    1  0.16752465D-02  0.14086188D-06  0.00000000D+00
    2  0.20435276D-02  0.12838572D-05  0.00000000D+00
    3 -0.20690747D-01 -0.71654588D-06  0.00000000D+00
    4  0.24306485D-02  0.15047496D-05  0.00000000D+00
    5  0.72779764D-02  0.23394016D-06  0.00000000D+00
    6  0.40794324D-02  0.24163474D-05  0.00000000D+00
    7  0.13043483D-01  0.43338715D-06  0.00000000D+00
    8 -0.85501699D-02 -0.51395912D-05  0.00000000D+00
```

IX. SCALING ALGORITHMS

Poorly scaled problems, i.e., problems in which the unknowns BETA and DELTA vary over several orders of magnitude, can cause least squares procedures difficulty. ODRPACK's scaling algorithms (discussed below) attempt to overcome these difficulties automatically, although it is preferable for the user to choose the units of the variable space so that the estimated parameters will have roughly the same magnitude [Dennis and Schnabel, 1983]. When the variables have roughly the same magnitude, the ODRPACK scaling algorithm will select scale values that are roughly equal, and the resulting computations will be the same (except for the effect of finite precision arithmetic) as an unscaled analysis, i.e., an analysis in which all of the scale values are set to one. If the user does not do this, the ODRPACK scaling algorithm will select varying scale values. This will not change the optimal solution, but it may affect the number of iterations required, or, in some cases, whether the algorithm is or is not successful.

Users may substitute their own scaling values using subroutine arguments SCLD and SCLB (see section VII.B).

IX.A BETA Scaling

ODRPACK chooses the scale values for the estimated BETAs as follows.

```
If some of the starting values of BETA are nonzero then
   let BETA_max = the largest absolute value of the nonzero starting values
                  of BETA, and
       BETA_min = the smallest absolute value of the nonzero starting
                  values of BETA.

   For K = 1 to NP do
      if BETA(K) = zero then
         scale_BETA(K) = ten/BETA_min
      else
         if LOG10(BETA_max)-LOG10(BETA_min) > one then
            scale_BETA(K) = one/ABS(BETA(K))
         else
            scale_BETA(K) = one/BETA_max.

If all of the starting values of BETA are zero then
   for K = 1 to NP do
       scale_BETA(K) = one.
```

Users may substitute their own BETA scaling values via subroutine argument SCLB.

IX.B DELTA Scaling

ODRPACK chooses scale values for the estimated errors in the independent variables, i.e., for the DELTAs, as follows.

```
For J = 1 to M do

   If some of the values of the Jth column of X are nonzero then
      let  X_max = the largest nonzero absolute value in the Jth column of
                   array X, and
           X_min = the smallest nonzero absolute value in the Jth column
                   of array X.

      For I = 1 to N do
         if X(I,J) = zero then
            scale_X(I,J) = ten/X_min
         else
            if LOG10(X_max)-LOG10(X_min) > one then
               scale_X(I,J) = one/ABS(X(I,J))
            else
               scale_X(I,J) = one/X_max.

   If all of the values of the Jth column of X are zero then
      For I = 1 to N do
            scale_X(I,J) = one
```

Users may substitute their own DELTA scaling values via subroutine argument SCLD.

X. EXTRACTING INFORMATION FROM THE WORK VECTORS

X.A Extracting Information from Vector WORK

Upon return from a call to ODRPACK, array WORK contains various values, some of which may be of interest to the user.

To extract information from WORK, the following declaration statement must be added to the user's program:

```
   INTEGER
  +   DELTAI,EPSI,
  +   WSSI,WSSDEI,WSSEPI,RVARI,
  +   PARTLI,SSTOLI,TAUFCI,EPSMAI,OLMAVI,
  +   FJACBI,FJACXI,XPLUSI,BETACI,BETASI,BETANI,DELTSI,
  +   DELTNI,DDELTI,FSI,FNI,SI,SSSI,SSI,SSFI,TI,TTI,TAUI,
  +   ALPHAI,VCVI,OMEGAI,YTI,UI,QRAUXI,WRK1I,SDI,RCONDI,
  +   ETAI,ACTRSI,PNORMI,PRERSI,RNORSI,
  +   LWKMN
```

where DELTAI through RNORSI are variables that indicate the starting locations within WORK of the stored values, and LWKMN is the minimum acceptable length of array WORK.

The appropriate values of DELTAI through RNORSI are obtained by invoking subroutine SWINF when using either of the single precision ODRPACK subroutines, SODR or SODRC, and by invoking DWINF when using either of the double precision subroutines, DODR or DODRC. The call statements for SWINF and DWINF have the same argument lists. To invoke either subroutine, use

```
   CALL <winf>
  +   (N,M,NP,
  +   DELTAI,EPSI,
  +   WSSI,WSSDEI,WSSEPI,RVARI,
  +   PARTLI,SSTOLI,TAUFCI,EPSMAI,OLMAVI,
  +   FJACBI,FJACXI,XPLUSI,BETACI,BETASI,BETANI,DELTSI,
  +   DELTNI,DDELTI,FSI,FNI,SI,SSSI,SSI,SSFI,TI,TTI,TAUI,
  +   ALPHAI,VCVI,OMEGAI,YTI,UI,QRAUXI,WRK1I,SEI,RCONDI,
  +   ETAI,ACTRSI,PNORMI,PRERSI,RNORSI,
  +   LWKMN)
```

where SWINF should be substituted for <winf> when using single precision subroutines SODR and SODRC, and DWINF should be substituted for <winf> when using double precision subroutines DODR and DODRC. The values of N, M and NP must be input to SIWINF and DIWINF with exactly the same values as were used in the original call to ODRPACK. (If possible, users should extract these declaration and call statements from online ODRPACK documentation to avoid typographical errors.)

In the following descriptions of the information returned in WORK, (*) indicates values that are likely to be of greatest interest.

(*) WORK(DELTAI) is the first element of the N by M matrix, DELTA, containing the estimated errors in the independent variables at the solution,

DELTA(I,J) = WORK(DELTAI-1+I+(J-1)*N)

for I=1,...,N & J=1,...,M.

(*) WORK(EPSI) is the first element of the N vector, EPSILON, containing the estimated errors in the dependent variables at the solution,

```
EPSILON(I) = WORK(EPSI-1+I)
```

for I=1,...,N.

(*) WORK(WSSI) is the weighted sum of the squared observation errors (eq.3) at the time the computations stopped, i.e.,

```
WORK(WSSI) = WORK(WSSDEI) + WORK(WSSEPI)
```

where WORK(WSSDEI) and WORK(WSSEPI) are defined below.

(*) WORK(WSSDEI) is the weighted sum of the squared DELTAs at the time the computations stopped, i.e.,

```
                N     M
WORK(WSSDEI) = SUM [ SUM ( W(I)*D(I,J)*DELTA(I,J) )**2 ] .
               I=1   J=1
```

(*) WORK(WSSEPI) is the weighted sum of the squared EPSILONs at the time the computations stopped, i.e.,

```
                N
WORK(WSSEPI) = SUM [ ( W(I)*EPSILON(I) )**2 ] .
               I=1
```

(*) WORK(RVARI) is the estimated residual variance at the time the computations stopped, i.e.,

```
               N
              SUM [ ( W(I)*R(I) )**2 ]
              I=1
WORK(RVARI) = ------------------------
                         DF
```

where DF is the degrees of freedom of the fit, i.e., the number of observations with nonzero weighted derivatives with respect to either BETA or DELTA minus the number of parameters being estimated.

WORK(PARTLI) is the value of the stopping tolerance used to detect parameter convergence.

WORK(SSTOLI) is the value of the stopping tolerance used to detect sum of squares convergence.

WORK(TAUFCI) is the value of the factor used to compute the initial trust region radius.

WORK(EPSMAI) is the value of machine precision, i.e., the smallest value e such that 1+e>1.

WORK(OLMAVI) is the average number of steps to obtain the Levenberg-Marquardt parameter.

WORK(FJACBI) is the first element of the N by NPP matrix, FJACB, containing the weighted derivative with respect to BETA, evaluated at the solution if the covariance matrix was computed, otherwise evaluated at the beginning of the last iteration,

FJACB(I,J) = WORK(FJACBI-1+I+(J-1)*N)

for I=1,...,N & J=1,...,NP.

WORK(FJACXI) is the first element of the N by M matrix, FJACX, containing the weighted derivative with respect to X, evaluated at the solution if the covariance matrix was computed, otherwise evaluated at the beginning of the last iteration,

FJACX(I,J) = WORK(FJACXI-1+I+(J-1)*N)

for I=1,...,N & J=1,...,M.

(*) WORK(XPLUSI) is the first element of the N by M matrix containing the final estimates of X, i.e.,
estimated< X > = observed< X > + estimated< DELTA >
computed using the final estimates of DELTA,

XPLUSD(I,J) = WORK(XPLUSI-1+I+(J-1)*N)

for I=1,...,N & J=1,...,M.

WORK(BETACI) is the first element of the NP vector, BETAC, containing the current working estimates of the unfixed subset of the function parameters,

BETAC(I) = WORK(BETACI-1+I)

for I=1,...,NP.

WORK(BETASI) is the first element of the NP vector, BETAS, containing the previous working estimates of the unfixed subset of the function parameters,

BETAS(I) = WORK(BETASI-1+I)

for I=1,...,NP.

WORK(BETANI) is the first element of the NP vector, BETAN, containing the new working estimates of the unfixed subset of the function parameters,

BETAN(I) = WORK(BETANI-1+I)

for I=1,...,NP.

WORK(DELTSI) is the first element of the N by M matrix, DELTAS, containing the previous working estimates of the errors in the independent variables,

DELTAS(I,J) = WORK(DELTASI-1+I+(J-1)*N)

for I=1,...,N & J=1,...,M.

WORK(DELTNI) is the first element of the N by M matrix, DELTAN, containing the new working estimates of the errors in the independent variables,

DELTAN(I,J) = WORK(DELTANI-1+I+(J-1)*N)

for I=1,...,N & J=1,...,M.

WORK(DDELTI) is the first element of the N by M matrix containing the weighted estimated errors in the independent variables, DDELTA = (W*D)**2 * DELTA,

DDELTA(I,J) = WORK(DDELTI-1+I+(J-1)*N)

for I=1,...,N & J=1,...,M.

WORK(FSI) is the first element of the N vector, FS, containing the saved weighted estimated errors in the dependent variable,

FS(I) = WORK(FSI-1+I)

for I=1,...,N.

(*) WORK(FNI) is the first element of the N vector, FN, containing the final estimates of Y = FN(X+DELTA;BETA), i.e., estimated< Y > = observed< Y > + estimated< EPSILON > computed using the final estimates of EPSILON,

FN(I) = WORK(FNI-1+I)

for I=1,...,N.

WORK(SI) is the first element of the NP vector, S, containing the step in the estimated function parameters,

S(I) = WORK(SI-1+I)

for I=1,...,NP.

(*) WORK(SSSI) is the first element of the N + N*M vector, SSS, containing the weighted errors at the solution,

SSS(I) = WORK(SSSI-1+I)

for I=1,...,N + N*M, where the first N elements contain the weighted EPSILONs,

W(I)*EPSILON(I) = WORK(SSSI-1+I)

for I=1,...,N and the next N*M elements contain the weighted DELTAs,

W(I)*D(I,J)*DELTA(I,J) = WORK(SSSI-1+I+J*N)

for I=1,...,N & J=1,...,M.

WORK(SSI) is the first element of the NP vector, SS, containing the scale of the estimated function parameters, SS(I) = WORK(SSI-1+I)

for I=1,...,NP.

WORK(SSFI) is the first element of the NP vector, SSF, containing the scale of each of the function parameters,

SSF(I) = WORK(SSFI-1+I)

for I=1,...,NP.

WORK(TI) is the first element of the N by M array, T, containing the step in the estimated errors in the independent variable,

T(I,J) = WORK(TI-1+I+(J-1)*N)

for I=1,...,N & J=1,...,M.

WORK(TTI) is the first element of the N by M array, TT, containing the scale of each the estimated errors in the independent variable,

TT(I,J) = WORK(TTI-1+I+(J-1)*N)

for I=1,...,N & J=1,...,M.

WORK(TAUI) is the trust region radius at the time the computations stopped.

WORK(ALPHAI) is the Levenberg-Marquardt parameter at the time the computations stopped.

(*) WORK(VCVI) is the first element of the covariance matrix of the NPP unfixed parameters, stored as an upper triangular matrix,

```
VCV(I,J) = WORK(VCVI-1+I+(J-1)*N)
VCV(J,I) = VCV(I,J)
```

for I=1,...,NPP & J=I,...,NPP. The covariance matrix is only computed when the third digit of JOB is zero, and when the solution is full rank.

The covariance matrix is defined as

VCV = RVAR * inv(trans(FJACB)*OMEGA*FJACB)

where

RVAR is the residual variance of the fit,

```
       1     N
RVAR = -- * SUM  (W(I)*R(I))**2
       DF   I=1
```

with DF the number of observations with nonzero weighted derivatives with respect to either BETA or DELTA minus the number of parameters actually estimated, FJACB is the derivative of FN(X(I,J)+DELTA(I,J);BETA) with respect to BETA, evalutated at the solution,

OMEGA is the diagonal matrix which has (I,I)th element

```
                         W(I)**2
OMEGA(I,I)  = ---------------------
                    M  FJACX(I,J)**2
              1 + SUM -------------
                  J=1    D(I,J)**2
```

with FJACX(I,J) the derivative of FN(X(I,J)+DELTA(I,J);BETA) with respect to DELTA(I,J), evalutated at the solution (for ordinary least squares, OMEGA(I,I) reduces to W(I)**2),

inv(.) indicates the inverse of the designated matrix, and

trans(.) indicates the transpose of the designated matrix.

Note that the covariance matrix is an approximation based on a linearization of the model in the neighborhood of the solution. The validity of the approximation depends on the nonlinearity of the model, the variance and distribution of the errors, and the data itself. Confidence regions and intervals computed using the variance covariance matrix are often acceptable, but can be very inaccurate in some cases. When reliable confidence intervals and regions are required, other more accurate, but more computationally expensive methods of constructing them should be used. (See, e.g., Boggs and Donaldson [1989], Donaldson and Schnabel [1987], Efron [1985], and Fuller [1987].)

WORK(OMEGAI) is the first element of the N vector

```
OMEGA(I)  = WORK(OMEGAI-1+I)

                     W(I)**2
          = ---------------------
                  M  FJACX(I,J)**2
            1 + SUM -------------
                J=1    D(I,J)**2
```

for I=1,...,N, computed at the solution if the covariance matrix was calculated.

WORK(YTI) is the first element of the N vector containing the diagonal elements of

```
YT(I)  = WORK(YTI-1+I)

       = -diag[sqrt(OMEGA(I),I=1,...,N]*(G1-V*inv(E)*D*G2)
```

for I=1,...,N.

WORK(UI) is the first element of the N vector, U, containing the approximate null vector for FJACB,

U(I) = WORK(UI-1+I)

for I=1,...,N.

WORK(QRAUXI) is the first element of the NP vector, QRAUX, required to recover the QR decomposition of FJACB,

QRAUX(I) = WORK(QRAUXI-1+I)

for I=1,...,NP.

WORK(WRK1I) is the first element of the N by M matrix, WRK1, required for work space,

WRK1(I,J) = WORK(WRK1I-1+I+(J-1)*N)

for I=1,...,N & J=1,...,M.

(*) WORK(SEI) is the first element of the NP vector containing the standard errors of the function parameters BETA, i.e., the square roots of the diagonal entries of the covariance matrix stored in WORK(VCVI) for the unfixed parameters and zero for the fixed parameters,

SE(I) = WORK(SEI-1+I)

for I=1,...,NP. The standard errors are only computed when the third digit of JOB is zero, and when the solution is full rank.

Note that the covariance matrix used to compute the standard errors is an approximation based on a linearization of the model in the neighborhood of the solution. The validity of the approximation depends on the nonlinearity of the model, the variance and distribution of the errors, and the data itself. Confidence intervals computed using the covariance matrix are often acceptable, but can be very inaccurate in some cases. When reliable confidence intervals and regions are required, other more accurate, but more computationally expensive methods of constructing them should be used. (See, e.g., Boggs and Donaldson [1989], Donaldson and Schnabel [1987], Efron [1985], and Fuller [1987].)

(*) WORK(RCONDI) is the reciprocal of the condition number at the time the computations stopped.

(*) WORK(ETAI) is the value of the relative error in the model function value.

WORK(ACTRSI) is the saved actual relative reduction in the weighted sum of squares of the observation errors from the last iteration.

WORK(PNROMI) is the norm of the scaled estimated parameters from the last iteration.

WORK(PRERSI) is the saved predicted relative reduction in the weighted sum of the squares of the observation errors from the last iteration.

WORK(RNORSI) is the norm of the saved weighted observation errors from the last iteration.

X.B Extracting Information from Vector IWORK

Upon return from a call to ODRPACK, array IWORK contains various values, some of which may be of interest to the user.

To extract information from IWORK, the following declaration statement must be added to the user's program

```
   INTEGER
  +   MSGB,MSGX,JPVTI,
  +   NNZWI,NPPI,IDFI,
  +   JOBI,IPRINI,LUNERI,LUNRPI,
  +   NROWI,NTOLI,NETAI,
  +   MAXITI,NITERI,NFEVI,NJEVI,INT2I,IRANKI,LDTTI,
  +   LIWKMN
```

where MSGB through LDTTI are variables that indicate the starting locations within IWORK of the stored values, and LIWKMN is the minimum acceptable length of array IWORK. The appropriate values of MSGB through LDTTI are obtained by invoking subroutine SIWINF when using either of the single precision ODRPACK subroutines, SODR or SODRC, and by invoking DIWINF when using either of the double precision subroutines, DODR or DODRC. The call statements for SIWINF and DIWINF have the same argument lists. To invoke either subroutine, use

```
   CALL <iwinf>
  +   (M,NP,
  +   MSGB,MSGX,JPVTI,
  +   NNZWI,NPPI,IDFI,
  +   JOBI,IPRINI,LUNERI,LUNRPI,
  +   NROWI,NTOLI,NETAI,
  +   MAXITI,NITERI,NFEVI,NJEVI,INT2I,IRANKI,LDTTI,
  +   LIWKMN)
```

where SIWINF should be substituted for <iwinf> when using single precision subroutines SODR and SODRC, and DIWINF should be substituted for <iwinf> when using double precision subroutines DODR and DODRC. Note that the values of M and NP must be input to SIWINF and DIWINF with exactly the same values as were used in the original call to ODRPACK. (If possible, users should extract these declaration and call statements from online ODRPACK documentation to avoid typographical errors.)

In the following descriptions of the information returned in IWORK, (*) indicates values that are likely to be of greatest interest.

(*) IWORK(MSGB) is the first element of the NP+1 vector, MSGB, used to indicate the results of checking the partial derivatives with respect to BETA.

The value of IWORK(MSGB) summarizes the results over all of the BETAs.

If IWORK(MSGB) < 0, the partial derivatives with respect to each of the BETAs were not checked.

If IWORK(MSGB) = 0, the partial derivatives with respect to each of the BETAs appear to be correct.

If IWORK(MSGB) = 1, the partial derivative with respect to at least one of the BETAs appears to be incorrect.

If IWORK(MSGB) = 2, the partial derivative with respect to at least one of the BETAs is questionable.

The value of IWORK(MSGB+K), K=1,...,NP, indicates the individual results for the partial derivative with respect to BETA(K), K=1,...,NP.

If IWORK(MSGB+K) = 0, the partial derivative with respect to BETA(K) appears to be correct.

If IWORK(MSGB+K) = 1, the partial derivative with respect to BETA(K) appears to be incorrect, i.e., the user supplied derivative and the finite difference value it is checked against do not agree to within the required tolerance and there is no reason to question the results.

If IWORK(MSGB+K) = 2, the partial derivative with respect to BETA(K) appears to be questionable because the user supplied derivative and the finite difference value it is checked against are both zero.

If IWORK(MSGB+K) = 3, the partial derivative with respect to BETA(K) appears to be questionable because the user supplied derivative is exactly zero and the finite difference value it is checked against is only approximately zero.

If IWORK(MSGB+K) = 4, the partial derivative with respect to BETA(K) appears to be questionable because the user supplied derivative is exactly zero and the finite difference value it is checked against is not even approximately zero.

If IWORK(MSGB+K) = 5, the partial derivative with respect to BETA(K) appears to be questionable because the finite difference value it is being checked against is questionable due to a high ratio of relative curvature to relative slope or to an incorrect scale value.

If IWORK(MSGB+K) = 6, the partial derivative with respect to BETA(K) appears to be questionable because the finite difference value it is being checked against is questionable due to a high ratio of relative curvature to relative slope.

(*) IWORK(MSGX) is the first element of the M+1 vector, MSGX, used to indicate the results of checking the partial derivatives with respect to X.

The value of IWORK(MSGX) summarizes the results over all of the Xs.

If IWORK(MSGX) < 0, the partial derivatives with respect to each of the Xs were not checked.

If IWORK(MSGX) = 0, the partial derivatives with respect to each of the Xs appear to be correct.

If IWORK(MSGX) = 1, the partial derivative with respect to at least one of the Xs appears to be incorrect.

If IWORK(MSGX) = 2, the partial derivative with respect to at least one of the Xs is questionable.

The value of IWORK(MSGX+J), J=1,...,M, indicates the individual results for the partial derivative with respect to the Jth column of X, J=1,...,M.

If IWORK(MSGX+J) = 0, the partial derivative with respect to the Jth column of X appears to be correct.

If IWORK(MSGX+J) = 1, the partial derivative with respect to the Jth column of X to be incorrect, i.e., the user supplied derivative and the finite difference value it is checked against do not agree to within the required tolerance and there is no reason to question the results.

If IWORK(MSGX+J) = 2, the partial derivative with respect to the Jth column of X appears to be questionable because the user supplied derivative and the finite difference value it is checked against are both zero.

If IWORK(MSGX+J) = 3, the partial derivative with respect to the Jth column of X appears to be questionable because the user supplied derivative is exactly zero and the finite difference value it is checked against is only approximately zero.

If IWORK(MSGX+J) = 4, the partial derivative with respect to the Jth column of X appears to be questionable because the user supplied derivative is exactly zero and the finite difference value it is checked against is not even approximately zero.

If IWORK(MSGX+J) = 5, the partial derivative with respect to the Jth column of X appears to be questionable because the finite difference value it is being checked against is questionable due to a high ratio of relative curvature to relative slope or to an incorrect scale value.

If IWORK(MSGX+J) = 6, the partial derivative with respect to the Jth column of X appears to be questionable because the finite difference value it is being checked against is questionable due to a high ratio of relative curvature to relative slope.

IWORK(JPVTI) is the first element of the NP vector, JPVT, containing the pivot vector,

JPVT(I) = WORK(JPVTI-1+I)

for I=1,...,NP.

IWORK(NNZWI) is the number of nonzero observation error weights.

IWORK(NPPI) is the number of function parameters actually being estimated.

(*) IWORK(IDFI) is the degrees of freedom of the fit, equal to the number of observations with nonzero weighted derivatives with respect to either BETA or DELTA minus the number of parameters being estimated.

IWORK(JOBI) is the value used to specify problem initialization and computational methods.

IWORK(IPRINI) is the print control value used.

IWORK(LUNERI) is the logical unit number used for error reports.

IWORK(LUNRPI) is the logical unit number used for computation reports.

IWORK(NROWI) is the number of the row at which the derivative is to be checked.

IWORK(NTOLI) is the number of digits of agreement required between the numerical derivatives and the user supplied derivatives.

(*) IWORK(NETAI) is the number of good digits in the model function results for the first row of the data not containing zero.

IWORK(MAXITI) is the maximum number of iterations allowed.

(*) IWORK(NITERI) is the number of iterations taken.

(*) IWORK(NFEVI) is the number of function evaluations made.

(*) IWORK(NJEVI) is the number of Jacobian matrix evaluations made.

IWORK(INT2I) is the number of internal doubling steps taken at the time the computations stopped.

(*) IWORK(IRANKI) is the rank deficiency at the solution.

IWORK(LDTTI) is the leading dimension of the work array TT.

XI. ACKNOWLEDGMENTS

The ODRPACK code was developed at the National Institute of Standards and Technology (formerly the National Bureau of Standards). The subroutine that supplies the value of machine precision was modeled after subroutines R1MACH and D1MACH from the Bell Laboratories "Framework for a Portable Library" [Fox et al., 1978]. We also use subroutines from LINPACK [Dongarra et al., 1979] and from the "Basic Linear Algebra Subprograms for Fortran Usage (BLAS)" [Lawson et al., 1979]. The code that checks user supplied derivatives was adapted from STARPAC [Donaldson and Tryon, 1986] using algorithms developed by Schnabel [1982].

XII. REFERENCES

Boggs, P. T., R. H. Byrd, J. R. Donaldson and R. B. Schnabel (1987a), "ODRPACK -- Software for Weighted Orthogonal Distance Regression," University of Colorado Department of Computer Science Technical Report Number CU-CS-360-87. (To appear in ACM Trans. Math. Software.)

Boggs, P. T., R. H. Byrd, and R. B. Schnabel (1987b), "A stable and efficient algorithm for nonlinear orthogonal distance regression," SIAM J. SCI. STAT. COMPUT., 8(6):1052-1078.

Boggs, P. T., and J. R. Donaldson (1989), "The Computation and Use of the Asymptotic Covariance Matrix for Measurement Error Models," National Institute of Standards and Technology Internal Report 89-4102.

Dennis, J. E., and R. B. Schnabel (1983), NUMERICAL METHODS FOR UNCONSTRAINED OPTIMIZATION AND NONLINEAR EQUATIONS, Prentice-Hall, Englewood Cliffs, NJ.

Donaldson, J. R., and R. B. Schnabel (1987), "Computational Experience with Confidence Regions and Confidence Intervals for Nonlinear Least Squares," TECHNOMETRICS, 29(1):67-82.

Donaldson, J. R., and P. V. Tryon (1986), "STARPAC - the standards time series and regression package," National Bureau of Standards (U.S.) Interim Report 86-3448.

Dongarra, J. J., C. B. Moler, J. R. Bunch, and G. W. Stewart (1979), LINPACK USERS' GUIDE, SIAM, Philadelphia, PA.

Draper, N. R., and H. Smith (1981), APPLIED REGRESSION ANALYSIS, Second Edition, John Wiley and Sons, New York, NY.

Efron, B. (1985), "The Jackknife, the Bootstrap and Other Resampling Plans," Monograph 38 (CBMS-NFS), SIAM, Philadelphia, PA.

Fox, P. A., A. D. Hall and N. L. Schryer (1978), "Algorithm 528: framework for a portable library [z]," ACM TRANS. MATH. SOFTWARE, 4(2):177-188.

Fuller, W. A. (1987), MEASUREMENT ERROR MODELS, John Wiley and Sons, New York, NY.

Gill, P. E., W. Murray and M. H. Wright (1981), PRACTICAL OPTIMIZATION, Academic Press, New York, NY.

Himmelblau, D. M. (1970), PROCESS ANALYSIS BY STATISTICAL METHODS, John Wiley and Sons, New York, NY.

Lawson, C., R. Hanson, D. Kincaid, and F. Krogh (1979), "Basic linear algebra subprograms for fortran usage", ACM TRANS. MATH. SOFTWARE, 5(3):308-371.

Schnabel, R. B. (1982), "Finite difference derivatives - theory and practice", (unpublished, available from author).

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM.

BIBLIOGRAPHIC DATA SHEET *(See instructions)*

1. PUBLICATION OR REPORT NO.
NISTIR 89-4103

2. Performing Organ. Report No.

3. Publication Date
AUGUST 1989

4. TITLE AND SUBTITLE

User's Reference Guide for ODRPACK Software for
Weighted Orthogonal Distance Regression, Version 1.7

5. AUTHOR(S)

Paul T. Boggs, Richard H. Byrd, Janet R. Donaldson and Robert B. Schnabel

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

10. SUPPLEMENTARY NOTES

□ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

ODRPACK is a portable collection of ANSI 77 Fortran subroutines for fitting a model to data. It is designed primarily for instances when the independent as well as the dependent variables have significant errors, implementing a highly efficient algorithm for solving the weighted orthogonal distance regression problem, i.e., for minimizing the sum of the squares of the weighted orthogonal distances between each data point and the curve described by the model equation. It can also be used to solve the ordinary least squares problem where all of the errors are attributed to the observations of the dependent variable. A complete description of the orthogonal distance regression problem and the algorithm implemented in ODRPACK is given by Boggs et al. [1987a and 1987b]. ODRPACK is designed to handle many levels of user sophistication and problem difficulty.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

orthogonal distance regression; measurement error models; nonlinear least squares, errors in variables.

13. AVAILABILITY

☒ Unlimited

□ For Official Distribution. Do Not Release to NTIS

□ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES
78

15. Price
$14.95